

Teste de INFORMATICĂ

pentru admiterea la
Universitatea Politehnica din București

București

2020

CUVÂNT ÎNAINTE

Ne bucurăm că începând cu acest an, Universitatea Politehnica din București (UPB) a adăugat informatica ca disciplină de admitere pentru studiile de licență. Am considerat că acest demers este important, în special, pentru facultățile care au specializări legate de Tehnologiile Informației și a Comunicațiilor (TIC). Totuși, admiterea la informatică a fost acceptată ca probă de admitere de majoritatea facultăților din UPB, un număr de 11 facultăți oferind posibilitatea elevilor să aleagă informatica ca probă de concurs în anul 2020.

Deși este o disciplină mult mai tânără comparativ cu celelalte probe de concurs, informatica a ajuns să fie un element esențial în educația inginerilor, chiar și a celor cu specializări din afara domeniului TIC. În lumea de astăzi, programarea calculatoarelor și gândirea algoritmică sunt aspecte de bază pentru rezolvarea de probleme practice într-o gamă variată de domenii. Mai mult, majoritatea domeniilor ingineresti necesită măcar minimal aplicarea unor concepte din informatică, pornind de la programare în diverse limbaje (precum C/C++, Java, Python, Matlab, R, SPSS etc.), unele generale și altele specifice anumitor domenii, și mergând până la înțelegerea și eventual proiectarea unor algoritmi particulari fiecărui domeniu de studiu.

Așadar, sperăm ca acest demers să fie de bun augur pentru întreaga comunitate UPB, dar și un exemplu pentru celelalte universități tehnice din România în promovarea informaticii ca disciplină de admitere la facultate. În acest mileniu, informatica va fi esențială pentru un număr din ce în ce mai mare de specialități tehnice.

În același timp, organizarea admiterii la informatică vine cu o responsabilitate suplimentară atât pentru comunitatea de profesori din cadrul UPB, cât și din ciclul preuniversitar. Astfel, profesorii din UPB care predau programare și algoritmi, dar și alte materii mai avansate, au trebuit să se familiarizeze cu terminologia și curricula de informatică din liceu. În acest context, ne-am bucurat de sprijinul și colaborarea cu o comunitate largă de profesori de informatică din ciclul preuniversitar. Aceasta a fost benefică pentru ambele comunități și ne dorim continuarea ei și în viitor, pentru a le oferi un instrument util elevilor de liceu pasionați de informatică și care își doresc să ajungă studenți în cadrul Universității Politehnica din București și, ulterior, ingineri. Un aspect observat în cadrul acestei colaborări a fost că deși în liceu se folosește notația „O” pentru analiza de complexitate a algoritmilor, aceasta are de fapt sensul notației „ θ ” și este folosită ca atare și în acest volum.

La final, în calitate de coordonator al comisiei de admitere la informatică în cadrul UPB, doresc să mulțumesc întregii echipe de colegi atât din ciclul preuniversitar, cât și din universitate, care au participat la elaborarea acestor variante de subiecte propuse pentru pregătirea candidaților înainte de concurs – lista completă este disponibilă în continuare. Mai mult, împreună cu colegii din universitate vom organiza sesiuni de pregătire a admiterii la informatică în lunile antemergătoare admiterii. Vă așteptăm în număr cât mai mare la concursul de admitere la informatică, unde împreună cu colegii din UPB vom avea sarcina dificilă, dar plină de recompense, de a propune setul de probleme pentru concurs. Pentru orice sugestii sau observații legate de acest volum, puteți să ne scrieți la adresa admitere.informatica@upb.ro.

Succes la pregătirea pentru examen!

Traian Rebedea
Conferențiar universitar
Facultatea de Automatică și Calculatoare

Penea Ștefania	Colegiul Național „Sfântul Sava”
Florea Andrei	Colegiul Național „I.L.Caragiale”
Rusu Oana	Liceul Greco-Catolic „Timotei Cipariu”
Bălașa Filonela	Colegiul Național „Grigore Moisil”
Crăciunescu Georgeta Antonia Rodica	Colegiul Național „Elena Cuza”
Săcuiu Silviu - Eugen	Colegiul Național „Mihai Viteazul”
Smîntîină Rodica	Colegiul Național „Gh. Șincai”
Balcă Mariana - Mihaela	Colegiul Național „Școala Centrală”
Anca Mihaela	Colegiul Național „I.L.Caragiale”
Badea Corina Elena	Liceul Teoretic „Al. I. Cuza”
Berbece Georgiana - Ligia	Liceul Teoretic „Nichita Stănescu”
Bușe Constanța Elena	Colegiul Național „Ion Neculce”
Chiriță Valentina	Liceul Teoretic „Al. I. Cuza”
Ciocaru Luminița	Liceul Teoretic „Dante Alighieri”
Danciu Alina	Colegiul Național „Ion Creangă”
Druță Doina Luminița	Liceul Teoretic „Dante Alighieri”
Dumitrescu Vasilica Iuliana	Colegiul Național „I.L.Caragiale”
Dumitru Silviu - Iulian	Colegiul Național „Gh. Lazăr”
Gebăilă Gilda - GrațIELA	Colegiul Național „Mihai Viteazul”
Manz Victor - Claudiu	Colegiul Național de Informatică „Tudor Vianu”
Mitrache Adrian	Liceul Teoretic „Eugen Lovinescu”
Mitrache Claudia Elena	Colegiul de Poștă și Telecomunicații „Gh. Airinei”
Petrișor Valiana Felicia	Colegiul Național Bilingv „George Coșbuc”
Popa Simona Mihaela	Colegiul Național „Gh. Lazăr”

Preda Doina Lavinia
Rusu Vicențiu

Rebedea Traian
Ablachim Denis
Chiroiu Mihai
Gliga Lavinus Ioan
Mocanu Irina
Niculescu Dragoș
Olteanu Alexandru
Pop Florin
Posea Vlad
Rușeți Ștefan
Sperilă Andrei
Trancă Cristian
Udrea Andreea

Liceul Teoretic „Dante Alighieri”
Liceul Greco-Catolic „Timotei Cipariu”

Universitatea Politehnica din București
Universitatea Politehnica din București
Universitatea Politehnica din București
Universitatea Politehnica din București
Universitatea Politehnica din București
Universitatea Politehnica din București
Universitatea Politehnica din București
Universitatea Politehnica din București
Universitatea Politehnica din București
Universitatea Politehnica din București
Universitatea Politehnica din București
Universitatea Politehnica din București
Universitatea Politehnica din București

PROGRAMA DE EXAMEN PENTRU DISCIPLINA INFORMATICĂ

Programa de examen pentru disciplina informatică, în cadrul admiterii la Facultatea de Automatică și Calculatoare, valabilă pentru anul 2020, urmează programa de la examenul de bacalaureat stabilită prin Anexa nr. 2 la OMECTS nr. 4800/31.VIII. 2010.

1. Algoritmi

- 1.1. Noțiunea de algoritm, caracteristici
- 1.2. Date, variabile, expresii, operații
- 1.3. Structuri de bază (liniară, alternativă și repetitivă)
- 1.4. Descrierea algoritmilor (programe pseudocod)

2. Elementele de bază ale unui limbaj de programare (Pascal sau C, la alegere)

- 2.1. Vocabularul limbajului
- 2.2. Constante. Identificatori
- 2.3. Noțiunea de tip de dată. Operatori aritmetici, logici, relaționali
- 2.4. Definirea tipurilor de date
- 2.5. Variabile. Declararea variabilelor
- 2.6. Definirea constantelor
- 2.7. Structura programelor. Comentarii
- 2.8. Expresii. Instrucțiunea de atribuire
- 2.9. Citirea/scrierea datelor
- 2.10. Structuri de control (instrucțiunea compusă, structuri alternative și repetitive)

3. Subprograme predefinite

- 3.1. Subprograme. Mecanisme de transfer prin intermediul parametrilor
- 3.2. Proceduri și funcții predefinite

4. Tipuri structurate de date

- 4.1. Tipul tablou
- 4.2. Tipul șir de caractere
 - operatori, proceduri și funcții predefinite pentru: citire, afișare, concatenare, căutare, extragere, inserare, eliminare și conversii (șir $\leftrightarrow$ valoare numerică)
- 4.3. Tipul înregistrare

5. Fișiere text

5.1. Fișiere text. Tipuri de acces

5.2. Proceduri și funcții predefinite pentru fișiere text

6. Algoritmi elementari

6.1. Probleme care operează asupra cifrelor unui număr

6.2. Divizibilitate. Numere prime. Algoritmul lui Euclid

6.3. Șirul lui Fibonacci. Calculul unor sume cu termenul general dat

6.4. Determinare minim/maxim

6.5. Metode de ordonare (metoda bulelor, inserției, selecției, numărării)

6.6. Interclasare

6.7. Metode de căutare (secvențială, binară)

6.8. Analiza complexității unui algoritm (considerând criteriile de eficiență *durata de executare* și *spațiu de memorie utilizat*)

7. Subprograme definite de utilizator

7.1. Proceduri și funcții

- declarare și apel

- parametri formali și parametri efectivi

- parametri transmiși prin valoare, parametri transmiși prin referință

- variabile globale și variabile locale, domeniu de vizibilitate

7.2. Proiectarea modulară a rezolvării unei probleme

8. Recursivitate

8.1. Prezentare generală

8.2. Proceduri și funcții recursive

9. Metoda backtracking (iterativă sau recursivă)

9.1. Prezentare generală

9.2. Probleme de generare. Oportunitatea utilizării metodei backtracking

10. Generarea elementelor combinatoriale

10.1. Permutări, aranjamente, combinări

10.2. Produs cartezian, submulțimi

11. Grafuri

11.1. Grafuri neorientate

- terminologie (nod/vârf, muchie, adiacență, incidență, grad, lanț, lanț elementar, ciclu, ciclu elementar, lungime, subgraf, graf parțial)

- proprietăți (conex, componentă conexă, graf complet, hamiltonian, eulerian)
- metode de reprezentare (matrice de adiacență, liste de adiacență)

11.2. Grafuri orientate

- terminologie (nod/vârf, arc, adiacență, incidență, grad intern și extern, drum, drum elementar, circuit, circuit elementar, lungime, subgraf, graf parțial)
- proprietăți (tare conexitate, componentă tare conexă)
- metode de reprezentare (matrice de adiacență, liste de adiacență)

11.3. Arbori

- terminologie (nod, muchie, rădăcină, descendent, descendent direct/fiu, ascendent, ascendent direct/părinte, frați, nod terminal, frunză)
- metode de reprezentare în memorie (matrice de adiacență, liste "de descendenți", vector "de tați")

VARIANTE

Varianta 1

1. Variabila **a** este de tip real. Pentru a verifica dacă valoarea variabilei **a** aparține mulțimii $[-3,2] \cup \{3, 5, 9\}$ se va utiliza următoarea expresie:

Limbajul C++/ Limbajul C

- a) `!((a<-3) || (a>2)) || (a==3) || (a==5) || (a==9)`
- b) `(a>=-3) && (a<=2) && (a==3) || (a==5) || (a==9)`
- c) `(a>-3) && (a<2) || (a==3) || (a==5) || (a==9)`
- d) `(a<-3) || (a>2) && (a==3) && (a==5) && (a==9)`
- e) `(a>=-3) || ((a<=2) && (a==3)) || (a==5) || (a==9)`
- f) `(a>=-3) || !((a>2)) || (a==3) || (a==5) || (a==9)`

Limbajul Pascal

- a) `not ((a<-3) or (a>2)) or (a=3) or (a=5) or (a=9)`
- b) `(a>=-3) and (a<=2) and (a=3) or (a=5) or (a=9)`
- c) `(a>-3) and (a<2) or (a=3) or (a=5) or (a=9)`
- d) `(a<-3) or (a>2) and (a=3) and (a=5) and (a=9)`
- e) `(a>=-3) or ((a<=2) and (a=3)) or (a=5) or (a=9)`
- f) `(a>=-3) or not((a>2)) or (a=3) or (a=5) or (a=9)`

2. Precizați cu ce expresie trebuie înlocuite punctele de suspensie, astfel încât în urma executării secvenței alăturate, să se deplaseze elementele $x_p, x_{p+1}, \dots, x_k$ ale unui tablou unidimensional **x**, cu **q-1** poziții spre dreapta.

Limbajul C++/ Limbajul C

```
for (j=k; j>=p; j--)
    x[...]=x[j];
```

- a) $q-1-j$ b) $j-q+1$ c) $q-1+j$ d) $q-2+j$ e) $q-j+1$ f) $j+q-3$

Limbajul Pascal

```
for j:= k downto p do
    x[...]:=x[j];
```

3. Precizați ce se va afișa după executarea secvenței de program de mai jos.

Limbajul C++/ Limbajul C

```
char a[20]="informatica", b[20]="";
strncat(b,a,strlen(strchr(a,'t')));
cout<<b; | printf("%s", b);
```

- a) tica b) form c) ica d) inf e) rmatica f) info

Limbajul Pascal

```
var b : string[20];
begin
b:='informatica';
delete(b,pos('r',b),pos('a',b));
write(b);
end.
```

4. Precizați care dintre următoarele secvențe calculează suma elementelor de pe linia **p**, ale unui tablou bidimensional **x**, cu **m** linii și **n** coloane (numerotate de la **1** la **m**, respectiv de la **1** la **n**)

Limbajul C++/ Limbajul C

```
a) s=0;
    for (i=m; i>=1; i--)
        s=s+x[p][i];

b) s=0; i=1;
    while(i<=m)
    { s=s+x[i][p];
      i++;}

c) s=0;
    for(i=n; i>=1; i--)
        s=s+x[i][p];

d) s=0; i=1;
    while(i<=n)
    {s=s+x[p][i];
     i++;}

e) s=0;
    for (i=m; i>1; i--)
        s=s+x[p][i];

f) s=0;
    for(i=m; i>=1; i--)
        s=s+x[i][p];
```

Limbajul Pascal

```
a) s:=0;
    for i:=m downto 1 do
        s:=s+x[p, i];

b) s:=0; i:=1;
    while i<=m do
    begin
        s:=s+x[i, p];
        i:=i+1;
    end;

c) s:=0;
    for i:=n downto 1 do
        s:=s+x[i, p];

d) s:=0; i:=1;
    while i<=n do
    begin
        s:=s+x[p, i];
        i:=i+1;
    end;

e) s:=0;
    for i:=m downto 2 do
        s:=s+x[p,i];

f) s:=0;
    for i:=m downto 1 do
        s:=s+x[i, p];
```

5. Fie graful orientat cu 5 noduri numerotate de la 1 la 5, și arcele (1,2), (2,3), (3,1), (3,4), (4,5), (5,4). Precizați care este matricea drumurilor asociată acestui graf. Matricea drumurilor este o matrice pătratică de dimensiune $n \times n$, definită astfel:

Limbajul C++/ Limbajul C

- $a[i][j]=1$ dacă există cel puțin un drum de la nodul i la nodul j
- $a[i][j]=0$ dacă nu există niciun drum de la nodul i la nodul j

```
a) 1 1 1 1 1
    1 1 1 1 1
    1 1 1 1 1
    0 0 0 1 1
    0 0 0 1 1
```

```
b) 0 1 1 1 1
    1 0 1 1 1
    1 1 0 1 1
    0 0 0 0 1
    0 0 0 1 0
```

```
d) 0 1 0 0 0
    0 0 1 0 0
    1 0 0 1 0
    0 0 0 0 1
    0 0 0 1 0
```

```
e) 1 1 1 1 1
    1 1 1 1 1
    1 1 1 1 1
    1 0 0 1 1
    0 1 0 1 1
```

Limbajul Pascal

- $a[i,j]=1$ dacă există cel puțin un drum de la nodul i la nodul j
- $a[i,j]=0$ dacă nu există niciun drum de la nodul i la nodul j

```
c) 1 1 1 1 1
    1 1 1 1 1
    1 1 1 1 1
    0 0 1 1 1
    0 0 0 1 1
```

```
f) 1 1 0 0 0
    0 0 1 0 0
    1 0 0 1 1
    0 1 0 0 1
    1 0 0 1 0
```

6. Utilizând metoda backtracking se generează elementele produsului cartezian a n mulțimi: $A_1, A_2, \dots, A_n$. Utilizând acest algoritm pentru a genera elementele produsului cartezian a 3

mulțimi: $A_1=\{1, 2, 3\}$, $A_2=\{1, 2\}$ și $A_3=\{1, 2, 3, 4\}$ atunci, precizați care din următoarele secvențe **nu** reprezintă o soluție a acestui algoritm, pentru produsul $A_3 \times A_2 \times A_1$.

- a) (4, 2, 3) b) (3, 3, 3) c) (3, 2, 1)
d) (1, 1, 1) e) (4, 1, 2) f) (3, 1, 3)

7. Fie funcția **p** definită mai jos

Limbajul C++/ Limbajul C

```
int p (int a, int b)
{if (b==0) return 0;
 else
  if (a%b==0) return p(a, b-1)+1;
  else return p(a, b-1);}
```

Limbajul Pascal

```
function p (a, b: integer) : integer;
begin
  if b=0 then p:=0
  else
    if a mod b=0 then p:=p(a, b-1)+1
    else p:=p(a,b-1);
end;
```

precizați care este apelul corect al funcției **p** pentru a verifica dacă un număr **x** este prim.

Limbajul C++/ Limbajul C

a) if (p(x,x)==2) cout<<"prim";	printf("prim");
b) if (p(2,x)==2) cout<<"prim";	printf("prim");
c) if (p(x,2)==0) cout<<"prim";	printf("prim");
d) if (p(x,x/2)==2) cout<<"prim";	printf("prim");
e) if (p(x,2)==x) cout<<"prim";	printf("prim");
f) if (p(2,x)==1) cout<<"prim";	printf("prim");

Limbajul Pascal

```
a) if p(x, x)=2 then write('prim');
b) if p(2, x)=2 then write('prim');
c) if p(x, 2)=0 then write('prim');
d) if p(x, x div 2)=2 then write('prim');
e) if p(x, 2)=x then write('prim');
f) if p(2, x)=1 then write('prim');
```

8. Un arbore are nodurile numerotate cu numere de la 1 la 5. Vectorul de tați asociat arborelui poate fi:

- a) 5, 4, 2, 1, 3 b) 2, 1, 0, 3, 4 c) 5, 2, 4, 5, 0
d) 2, 4, 0, 3, 4 e) 0, 2, 4, 5, 0 f) 1, 4, 0, 3, 4

9. Se consideră subprogramul **t** având definiția următoare:

Limbajul C++

```
void t(int &x, int y)
{ x=x-1; y=x+1;
  cout<<x<<y;
}
```

Limbajul C

```
void t(int *x, int y)
{ *x=*x-1; y=*x+1;
  printf( "%d%d",*x,y);
}
```

Limbajul Pascal

```
procedure t( var x: integer; y: integer);
begin
  x:=x-1; y:=x+1;
  write(x, y);
end;
```

Dacă inițial $x=3$ și $y=7$, precizați ce se va afișa în urma executării secvenței de instrucțiuni:

Limbajul C++
`t(y,y);`
`cout<<x<<y;`
`t(y,x);`

Limbajul C
`t(&y,y);`
`printf( "%d%d",x,y);`
`t(&y,x);`

Limbajul Pascal
`t(y,y);`
`write(x,y);`
`t(y,x);`

a) 673556 b) 676767 c) 673767 d) 768978 e) 656376 f) 673656

- 10 Funcția **par** cu parametrii de tip întreg **a** și **b** determină numărul de valori pare din intervalul **[a, b]**, ($a < b$). Precizați care din următoarele expresii este adevărată, pentru orice numere **a** și **b** care **nu** au aceeași paritate.

Limbajul C++/ Limbajul C
a) `par(a, b) == b-a`
b) `par(a, b) == (b-a-1)/2`
c) `par(a, b) == (b-a+1)/2`
d) `par(a, b) == par(a, b +1)`
e) `par(a, b) == (b-a)/2`
f) `par(a, b) == par(a+1, b)`

Limbajul Pascal
a) `par(a, b) = b-a`
b) `par(a, b) = (b-a-1) DIV 2`
c) `par(a, b) = (b-a+1) DIV 2`
d) `par(a, b) = par(a, b +1)`
e) `par(a, b) = (b-a) DIV 2`
f) `par(a, b) = par(a+1, b)`

- 11 Fie antetul funcției **mini**:

Limbajul C++/ Limbajul C
`int mini (int x, int y)`

Limbajul Pascal
`function mini(x,y : integer): integer;`

care returnează minimul dintre valorile variabilelor **x** și **y**. Precizați instrucțiunea prin care se înlocuiesc punctele de suspensie, astfel încât la finalul executării secvenței de mai jos, să se afișeze minimul dintre elementele tabloului unidimensional **v**, care are 30 de elemente întregi aflate pe pozițiile 1, 2,...,30.

Limbajul C++/ Limbajul C
`c = mini(v[1], v[2]);`
`for(i=3; i<=30; i++)`
`.....`
`cout<<c;      |      printf( "%d",c);`

a) `c =mini(v[i], v[i+1]);`
b) `c =mini(mini(v[i],v[1]),v[i+1]);`
c) `c =mini(c, v[i]);`
d) `c =mini(v[1], v[i]);`
e) `c =mini(v[i], v[i-1]);`
f) `c =mini(c, v[30]);`

Limbajul Pascal
`c :=mini(v[1],v[2]);`
`for i:=3 to 30 do`
`.....`
`writeln(c);`

a) `c:=mini(v[i], v[i+1]);`
b) `c:=mini(mini(v[i],v[1]),v[i+1]);`
c) `c:=mini(c, v[i]);`
d) `c:=mini(v[1], v[i]);`
e) `c:=mini(v[i], v[i-1]);`
f) `c:=mini(c, v[30]);`

- 12 Precizați ce va conține variabila **s** după executarea următoarei secvențe de instrucțiuni.

Limbajul C++/ Limbajul C
`char s[30]="AdmiTerE";`
`int i;`

Limbajul Pascal
`var s: string[29];`
`i: integer;`

```

for (i=0; i<strlen(s); i++)
    if (s[i]>='A' && s[i]<='Z')
        s[i]=s[i]+32;
    else
        strcpy(s+i, s+i+1);
cout<<s;    | printf("%s", s);

```

```

.....
s:='AdmiTerE';
for i:=1 to length(s) do
    if s[i] in ['A'..'Z'] then
        s[i] := chr(ord(s[i])+32)
    else
        delete(s,i,1);
write(s);

```

a) ate b) amtre c) amre d) aTe e) amTre f)are

13 Pentru următoarele declarații:

Limbajul C++/ Limbajul C

```

typedef struct {
    char nume[20];
    int nr;
    int nota[15];
} elev;
elev e[28], x;

```

Limbajul Pascal

```

type elev = record
    nume : string[19];
    nr : integer;
    nota : array[1..14] of integer;
end;
var e: array[1..27] of elev;
x: elev;

```

precizați care dintre următoarele instrucțiuni este corectă din punct de vedere sintactic.

Limbajul C++/ Limbajul C

a) e[10] = x;
b) e[10] = x.nr;
c) e[10] = e.nota[10];
d) elev. nota[5] = 7;
e) x.nota = x.nota+1;
f) x.nr=x.nota;

Limbajul Pascal

a) e[10] := x;
b) e[10] := x.nr;
c) e[10] := e.nota[10];
d) elev. nota[5] := 7;
e) x. nota := x. nota +1;
f) x.nr := x.nota;

14 Precizați câte grafuri neorientate distincte, cu 6 noduri, numerotate de la 1 la 6, se pot construi, astfel încât nodul 2 să aibă gradul 1. Două grafuri sunt distincte dacă matricele lor de adiacență sunt diferite.

a) 46 b) 256 c) 6! d) 1024 e) 2345 f) 5120

15 Pentru funcția **g** definită mai jos, precizați valoarea care se returnează în urma apelului **g(2,1)**.

Limbajul C++/ Limbajul C

```

int g(int x, int y)
{ if (x > 0)
    { if (y == 0) return g(x-1,1);
      if (y > 0) return g(x-1, g(x, y-1));
    }
return y+1;
}

```

Limbajul Pascal

```

function g(x,y: integer): integer;
begin
    if x>0 then begin
        if y=0 then g:=g(x-1,1);
        if y>0 then g:=g(x-1,g(x,y-1));
    end
    else g:=y+1;
end;

```

a) 3 b) 4 c) 5 d) 6 e) 7 f) 10

Varianta 2

1. Precizați ce se va afișa pe ecran în urma executării următoarelor instrucțiuni.

Limbajul C++/ Limbajul C

```
for(i=1; i<=5; i++)
  for(j=5; j>=i; j--)
    if (i%2==0) cout<<i;
        | printf("%d",i);
    else cout<<j;
        | printf("%d",j);
```

Limbajul Pascal

```
for i :=1 to 5 do
  for j :=5 downto i do
    if i mod 2 = 0 then write(i)
                        else write(j);
```

- a) 12345 b) 111115432333545 c) 543212222543445
d) 122333444455555 e) 12334445555 f) 23344455554443

2. Se consideră expresia:

Limbajul C++/ Limbajul C

```
!((x<=y && x>=z) || x<=t)
```

Limbajul Pascal

```
not((x<=y) and (x>=z)) or (x<=t))
```

Precizați care expresie este echivalentă cu expresia dată.

Limbajul C++/ Limbajul C

- a) $x \leq y \parallel x \geq z \ \&\& \ x \leq t$
b) $x > y \parallel x < z \ \&\& \ x > t$
c) $x > y \ \&\& \ x < z \parallel x > t$
d) $(x > y \parallel x < z) \ \&\& \ x > t$
e) $x > y \parallel x < z \ \&\& \ x \leq t$
f) $x \geq y \ \&\& \ x \leq z \parallel x > t$

Limbajul Pascal

- a) $(x \leq y) \text{ or } (x \geq z) \text{ and } (x \leq t)$
b) $(x > y) \text{ or } (x < z) \text{ and } (x > t)$
c) $(x > y) \text{ and } (x < z) \text{ or } (x > t)$
d) $((x > y) \text{ or } (x < z)) \text{ and } (x > t)$
e) $(x > y) \text{ or } (x < z) \text{ and } (x \leq t)$
f) $(x \geq y) \text{ and } (x \leq z) \text{ or } (x > t)$

3. Un tablou bidimensional **a**, cu n linii și n coloane numerotate de la **1** la **n**, este simetric față de diagonala secundară dacă pentru orice pereche de indici (i, j) este adevărată expresia:

Limbajul C++/ Limbajul C

- a) $a[i][j] == a[j][i]$
b) $a[i][i] == a[n+1-j][n+1+i]$
c) $a[i][j] == a[n+1-i][n+1-j]$
d) $a[i][j] == a[n-j][n-i]$
e) $a[i][j] == a[n+1-j][n+1-i]$
f) $a[i][j] == a[n+j][n-i]$

Limbajul Pascal

- a) $a[i, j] = a[j, i]$
b) $a[i, i] = a[n+1-j, n+1+i]$
c) $a[i, j] = a[n+1-i, n+1-j]$
d) $a[i, j] = a[n-j, n-i]$
e) $a[i, j] = a[n+1-j, n+1-i]$
f) $a[i, j] = a[n+j, n-i]$

4. Precizați ce valoare are variabila **b** de tip șir de caractere după executarea secvenței:

Limbajul C++/ Limbajul C

```
char b[ ]="toc2019";
```

Limbajul Pascal

```
var b: string[30];
```

```
b[3]=b[3]-1;
strcpy(b+5,b+7);
strcpy(b,b+3);
```

```
b:= 'toc2019';
b[4]:= chr(ord(b[4])-1);
delete(b,6,2);
delete(b,1,3);
```

a) b20 b) 19 c) 20 d) 10 e) toc f)2019

5. Se consideră tipul de date **punct**, ce memorează abscisa și ordonata unui punct din plan și tipul de date **segment** ce memorează două puncte distincte reprezentând extremitățile unui segment din plan. Precizați care dintre următoarele expresii are ca valoare ordonata mijlocului segmentului corespunzător variabilei **s** de tip **segment**.

Limbajul C++/ Limbajul C

```
typedef struct
{ float x, y;}punct;
typedef struct
{punct A, B;}segment;
segment s;
```

Limbajul Pascal

```
type punct=record
x, y: real;
end;
type segment=record
A, B: punct;
end;
var s: segment;
```

a) $(A.x+B.x)/2$ b) $(A.s.y+B.s.y)/2$ c) $(s.y+s.y)/2$
d) $(A.x+B.y)/2$ e) $(s.A.x+s.B.x)/2$ f) $(s.A.y+s.B.y)/2$

6. Utilizând metoda backtracking se generează toate tablourile bidimensionale pătratice de ordin **n** ale căror elemente aparțin mulțimii $\{0, 1\}$, cu proprietatea că pe fiecare linie și pe fiecare coloană există **o singură** valoare 1. Dacă **n=3** tablourile bidimensionale sunt generate în ordinea următoare:

100	100	010	010	001	001
010	001	100	001	100	010
001	010	001	100	010	100

Dacă **n =4**, precizați care este tabloul bidimensional generat imediat după tabloul bidimensional:

```
0010
1000
0001
0100.
```

a) 0010 b) 0010 c) 0001 d) 0010 e) 0001 f) 1000
1000 0100 1000 0001 0010 0010
0100 1000 0010 1000 0100 0100
0001 0001 0100 0100 1000 0001

7. Pentru definiția de mai jos a subprogramului **nr**, stabiliți ce valoare returnează apelul **nr(6,3)**.

Limbajul C++/ Limbajul C

```
int nr (int x, int y)
```

Limbajul Pascal

```
function nr(x,y:integer): integer;
```

```

{ int a, t;
  if (x==y || y==1) return 1;
  if (x<y) return 0;
  a=0;
  for(t=1; t<=y; t++)
    a=a+nr(x-y,t);
  return a;
}

```

```

var a, t : integer;
begin
  if (x=y) or (y=1) then nr := 1
  else if x<y then nr :=0
    else begin
      a :=0;
      for t :=1 to y do
        a :=a+nr(x-y,t);
      nr := a;
    end;
end;

```

- a) 0 b) 1 c) 2 d) 3 e) 4 f) 6

8. Se consideră un arbore cu 10 noduri, numerotate de la 1 la 10 având vectorul de tați următor (0, 1, 1, 1, 3, 3, 3, 4, 7, 7). Descendenții nodului 3 sunt:

- a) 5, 6, 7 b) 5, 6, 7, 9, 10 c) 4, 5, 6, 7, 8, 9, 10
d) 6, 7 e) 4, 5, 6 f) 4, 7

9. Știind că inițial variabilele întregi **x** și **y** au valorile 1 și respectiv 2, stabiliți care sunt valorile lor după apelul **F(x,y)** (pentru variantele în limbajele C++ sau Pascal) respectiv **F(&x,y)** (pentru varianta în limbajul C):

Limbajul C++
void F(int &x, int y)
{
 x=2; y=3;
}

Limbajul C
void F(int *x, int y)
{
 ***x=2; y=3;**
}

Limbajul Pascal
procedure F(var x:
integer; y: integer);
begin
 x:=2; y:=3;
end;

- a) 3 2 b) 1 2 c) 1 1 d) 2 3 e) 2 2 f) 3 3

10. Se consideră un graf neorientat cu 8 noduri numerotate de la 1 la 8 și următoarele muchii: [1,7], [1,8], [3,4], [3,5], [3,6], [3,7], [4,7], [5,6], [5,8], [6,7], [6,8], [7,8]. Precizați care este numărul minim de culori cu care pot fi colorate nodurile grafului, astfel încât oricare două noduri adiacente să aibă culori diferite.

- a) 1 b) 2 c) 3 d) 4 e) 6 f) 8

11. Numărul maxim de muchii dintr-un graf neorientat cu 16 noduri și 7 componente conexe este:

- a) 15 b) 18 c) 23 d) 25 e) 36 f) 45

12. Se consideră un șir de caractere **c** de lungime maximă 20, ce conține cel puțin un caracter 'd'. Precizați care dintre următoarele secvențe afișează poziția primei apariții a lui 'd' în șirul de caractere **c**.

Limbajul C++/ Limbajul C

- a) **cout<<strchr(c, 'd')-c;** | **printf("%d", strchr(c, 'd')-c);**
b) **cout<<strrchr(c, 'd');** | **printf("%d", strrchr(c, 'd'));**
c) **cout<<strchr(c, 'd')-c-1;** | **printf("%d", strchr(c, 'd')-c-1);**
d) **cout<<strchr(c, 'd');** | **printf("%d", strchr(c, 'd'));**
e) **cout<<strchr(c, 'd')-c+2;** | **printf("%d", strchr(c, 'd')-c+2);**
f) **cout<<strchr(c, 'd-c');** | **printf("%d", strchr(c, 'd-c'));**

Limbajul Pascal

```
a) write(pos('d', c));
b) write(pos(c, 'd'));
c) write(pos(c, 'd')-1);
d) write(substr(c, 'd'));
e) write(pos(c, 'd')+2);
f) write(substr(c, 'd-c'));
```

- 13 Precizați ce valoare are variabila de tip întreg **a** după executarea următoarei secvențe de instrucțiuni.

Limbajul C++/ Limbajul C

```
a=2019;
for(x=1; x<=5; x++);
a=a+2;
```

Limbajul Pascal

```
a: = 2019;
for x:=1 to 5 do;
a: =a+2;
```

- a) 2019 b) 2020 c) 2021 d) 2024 e) 2027 f) 2029

- 14 O clasă de 30 de elevi este la ora de informatică și profesorul dorește să formeze o echipă de 5 elevi. El îi cere unui elev să-i genereze toate posibilitățile de a forma o grupă de 5 elevi din acea clasă. Această problemă este similară cu generarea tuturor:

- a) elementelor produsului cartezian A^5 , A fiind o mulțime cu 30 de elemente
- b) partițiilor unei mulțimi
- c) aranjamentelor de 30 de elemente luate câte 5
- d) permutărilor de 5 elemente
- e) combinațiilor de 30 de elemente luate câte 5
- f) submulțimilor de 5 elemente din mulțimea A, A fiind o mulțime cu 30 de elemente

- 15 Se consideră următoarea funcție recursivă:

Limbajul C++/ Limbajul C

```
int s(int t)
{ if (t == 1) return 0;
  else
    if (t == 2) return 1;
    else
      return s(t-2) + s(t-1); }
```

Limbajul Pascal

```
function s(t : integer) : integer;
begin
  if t = 1 then s := 0
  else
    if t = 2 then s := 1
    else
      s:=s(t-2)+s(t-1);
end;
```

Dacă apelul inițial **nu** se ia în considerare, precizați câte autoapeluri vor fi făcute pentru apelul s(6).

- a) 4 b) 6 c) 8 d) 10 e) 11 f) 14

Varianta 3

1. Precizați pentru câte valori naturale citite pentru variabila **n** programul următor afișează valoarea **10**.

```

Limbajul C++
#include<iostream>
using namespace std;
int main()
{
    int n,i=1,k=1;
    cin>>n;

    while(k*k<=n)k+=++i;
    cout<<k;
}

```

a) 0 b) 10

```

Limbajul C
#include<stdio.h>
void main( )
{
    int n,i=1,k=1;
    scanf("%d", &n);
    while(k*k<=n)
        k+=++i;
    printf("%d", k);
}

```

c) 54 d) 63

```

Limbajul Pascal
var n,i,k:integer;
begin
    i:=1; k:=1;
    readln(n);
    while k*k<=n do
        begin
            inc(i); k:=k+i;
        end;
    write(k)
end.

```

e) 64 f) 100

2. Se consideră următorul program:

```

Limbajul C++
#include<iostream>
using namespace std;
int main()
{
    int i, v[10];
    for(i=1;i<10;i++)
        cin>>v[i];
    i=v[7];
    do
    {
        cout<<i<<" ";
        i=v[i];
    }while(i!=1);
    cout<<i;
}

```

```

Limbajul C
#include<stdio.h>
void main()
{
    int i,v[10];
    for(i=1;i<10;i++)
        scanf("%d",&v[i]);
    i=v[7];
    do
    {
        printf("%d ",i);
        i=v[i];
    }while(i!=1);
    printf("%d ",i);
}

```

```

Limbajul Pascal
var i:integer;
    v:array[0..9]of integer;
begin
    for i:=1 to 9 do
        read(v[i]);
    i:=v[7];
    repeat
        write( i, ' ');
        i:=v[i];
    until i=1;
    write(i)
end.

```

La executarea programului se introduc, în ordine, valorile: 5 3 7 6 9 2 1 4 8.

Precizați care sunt valorile afișate.

- a) 1 5 9 8 4 6 2 3 7 1 b) 1 2 3 4 5 6 7 8 9 1
 c) 1 5 9 8 4 6 2 3 7 d) 1 5 3 7 6 9 2 4 8 1
 e) 7 1 2 9 8 4 6 2 3 1 f) 7 3 5 4 2 8 1 9 6 1

3. Precizați ce valoare se afișează în urma execuției următorului program dacă pentru **n** se citește valoarea **20**.

```

Limbajul C++
#include<iostream>
using namespace std;
int main()
{
    int a[51][51];
    int i,j,n, s=0;
    cin>>n;
    for(i=1;i<=n;i++)

```

```

Limbajul C
#include<stdio.h>
void main()
{
    int a[51][51];
    int i,j,n, s=0;
    scanf("%d", &n);
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)

```

```

Limbajul Pascal
var a:array[1..50,1..50]
    of integer;
    i,j,n,s: integer;
begin
    s:=0;
    readln(n);
    for i:=1 to n do

```

<pre>for(j=1;j<=n;j++) a[i][j]=i-j; for(i=1;i<=n;i++) for(j=1;j<=n;j++) if(i+1!=j) s+=a[i][j]; cout<<s; }</pre>	<pre> a[i][j]=i-j; for(i=1;i<=n;i++) for(j=1;j<=n;j++) if(i+1!=j) s+=a[i][j]; printf("%d ", s); }</pre>	<pre>for j:=1 to n do a[i,j]:=i-j; for i:=1 to n do for j:=1 to n do if i+1<> j then s:=s+ a[i,j]; write(s) end.</pre>			
a) -19	b) 0	c) 9	d) 19	e) 20	f) 190

4. Următoarea secvență de program folosește metoda căutării binare pentru a verifica dacă valoarea **x** se află printre cele **10** elemente ale tabloului unidimensional **v** (cu indicii de la 1 la 10), dar produce eroare de execuție pentru mai multe seturi de date.

Limbajul C++/ Limbajul C

```

i=1; j=10;
do {
    k=(i+j)/2;
    if(v[k]<x) i=k;
    else j=k;
} while(v[k]!=x&& i<j);
if(v[k]==x) cout<<"EXISTA";
    | printf("EXISTA");
else cout<<"NU EXISTA";
    | printf("NU EXISTA");

```

Limbajul Pascal

```

i:=1; j:=10;
repeat
    k:=(i+j)div 2;
    if v[k]<x then i:=k
    else j:=k
until (v[k]=x) or (i>=j);
if v[k]=x then write('EXISTA')
else write('NU EXISTA');

```

Precizați pentru care dintre seturile de date de intrare (elementele tabloului unidimensional **v** și valoarea lui **x**) secvența dată **nu** produce eroare de execuție.

- a) **v**=(1,2,3,4,5,6,7,8,9,10) și **x**>10
b) **v**=(2,4,6,8,10,12,14,16,18,20) și **x**>20
c) **v**=(2,2,2,2,2,2,2,2,2,2) și **x**>2
d) **v**=(2,4,6,8,10,12,14,16,18,20) și 2<**x**<20, **x** număr par
e) **v**=(1,2,5,4,3,6,7,8,9,10) și **x**=4
f) **v**=(2,4,6,8,10,12,14,16,18,20) și 2<**x**<20, **x** număr impar
5. Precizați câte elemente divizibile cu **10**, se vor afișa în urma executării programului următor.

Limbajul C++

```

#include<iostream>
using namespace std;
int f(int &y, int x)
{
    y=y/10+x;
    return x+y;
}
int main( )
{
    int x=101,y=10;
    cout<<f(x,y)<<" ";
    cout<<x<<" "<<y<<"
";
    cout<<f(x,y);
}

```

Limbajul C

```

#include<stdio.h>
int f(int *y, int x)
{
    *y=*y/10+x;
    return x+*y;
}
void main( )
{
    int x=101,y=10;
    printf("%d ",
f(&x,y));
    printf("%d %d ", x,
y);
    printf("%d ",
f(&x,y));
}

```

Limbajul Pascal

```

var x,y:integer;
function f(var
y:integer;
x:integer):integer;
begin
    y:=y div 10 + 10;
    f:=x + y
end;
begin
    x:=101; y:=10;
    write(f(x,y),' ');
    write(x,' ',y,' ');
    write(f(x,y))
end.

```

- a) 0 b) 1 c) 2 d) 3 e) 4 f) 5

6. Se consideră funcția:

Limbajul C++/ Limbajul C

```
unsigned f(unsigned x, unsigned
y)
{
    if(x==1) return y;
    if(x%2==0) return f(x/2,y*2);
    return y+f(x/2,y*2);
}
```

Limbajul Pascal

```
function f(x, y:word):word;
begin
    if x=1 then f:=y
    else
        if x mod 2=0 then f:=f(x div 2,
y*2)
        else f:=y + f(x div 2, y*2)
    end;
```

Precizați care dintre funcțiile următoare **nu** este echivalentă cu funcția dată.

Limbajul C++/ Limbajul C

a) unsigned f(unsigned x, unsigned y)

```
{
    if(x==0) return 0;
    if(x%2==0) return 2*f(x/2,y);
    return y+f(x-1,y);
}
```

Limbajul Pascal

a) function f(x,y:word):word;

```
begin
    if x=0 then f:=0
    else
        if x mod 2 = 0 then
            f:=2*f(x div 2, y)
        else f:=y + f(x-1, y)
    end;
```

b) unsigned f(unsigned x, unsigned y)

```
{
    if(x==0) return y;
    if(x%2==0) return f(x/2,y);
    return y+f(x-1,y);
}
```

b) function f(x,y:word):word;

```
begin
    if x=0 then f:=y
    else
        if x mod 2 = 0 then
            f:=f(x div 2, y)
        else f:=y + f(x-1, y)
    end;
```

c) unsigned f(unsigned x, unsigned y)

```
{
    return y*x;
}
```

c) function f(x,y:word):word;

```
begin
    f:=y*x
end;
```

d) unsigned f(unsigned x, unsigned y)

```
{
    if(x==0) return 0;
    return y+f(x-1,y);
}
```

d) function f(x,y:word):word;

```
begin
    if x=0 then f:=0
    else f:=y+f(x-1, y)
end;
```

e) unsigned f(unsigned x, unsigned y)

```
{
    unsigned s=0;
    while(x>0)
    {
        s=s+y;
        x--;
    }
    return s;
}
```

e) function f(x,y:word):word;

```
var s:word;
begin
    s:=0;
    while x>0 do
    begin
        s:=s+y; x:=x-1
    end;
    f:=s
end;
```

```
f) unsigned f(unsigned x, unsigned y)
{
    unsigned i,s=0;
    for(i=1;i<=y;i++)s=s+x;
    return s;
}
```

```
f) function f(x,y:word):word;
    var i,s:word;
    begin
        s:=0;
        for i:=1 to y do s:=s+x;
        f:=s
    end;
```

7. Pentru implementarea unei stive se definește structura următoare, în care câmpul **v** este un tablou unidimensional în care sunt memorate valorile din stivă, iar câmpul **k** reprezintă vârful stivei:

Limbajul C++/ Limbajul C

```
typedef struct
{
    int k ; int v[100];
}stiva;
```

Limbajul Pascal

```
type stiva=record
    k: integer;
    v: array[0..99] of integer;
end;
```

Variabila **s** este de tipul **stiva**, iar **x** este un număr întreg.

Precizați care dintre următoarele secvențe determină adăugarea corectă a valorii **x** în stivă.

Limbajul C++/ Limbajul C

- a) if (s.k<99) s.v[++s.k]=x;
- b) k++;v[k]=x;
- c) if (s.k<99) s.v[s.k]=x;
- d) k++; s.v[s.k]=x;
- e) s.v[s.k+1]:=x;
- f) x:=s.v[s.k];

Limbajul Pascal

- a) if s.k<99 then
 begin
 s.v[s.k+1]:=x;
 inc(s.k)
 end;
- b) inc(k); v[k]:=x;
- c) if s.k<99 then s.v[s.k]:=x;
- d) with s do begin
 k:=k+1; v[k]:=x
 end;
- e) s.v[s.k+1]:=x;
- f) x:=s.v[s.k];

8. Se consideră definit un subprogram care determină ștergerea tuturor aparițiilor unui caracter **c** din șirul de caractere **s**. Antetul subprogramului este:

(C++/C) void del (char s[255], char c);

(Pascal) procedure del (var s:string; c:character);

Precizați ce secvență poate fi utilizată pentru ștergerea tuturor caracterelor ce reprezintă cifre din șirul **s**. (**s** este un șir de caractere de lungime maxim **255**, iar **i** este o variabilă de tip caracter)

Limbajul C++/ Limbajul C

- a) del(s,"0123456789");
- b) del(s,i);
- c) for(i='0';i<='9';i++)del(i,s);
- d) for(i=0;i<=9;i++)del(s,i);
- e) for(i='0';i<='10';i++)del(s,i);
- f) for(i='0';i<='9';i++)del(s,i);

Limbajul Pascal

- a) del(s, '0123456789');
- b) del(s,i);
- c) for i:='0' to '9' do del(i,s);
- d) for i:= 0 to 9 do del(s,i);
- e) for i:='0' to '10' do del(s,i);
- f) for i:='0' to '9' do del(s,i);

9. Variabila **n** reprezintă un număr natural cu cel mult 3 cifre. Precizați pentru câte valori ale variabilei **n** expresia:

(C++/C) n/100+n%100/10+n%10

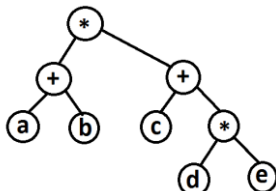
(Pascal) n div 100 + n mod 100 div 10 + n mod 10

are valoarea **9**.

- a) 9 b) 10 c) 45 d) 54 e) 55 f) 100

10. Unei expresii algebrice i se asociază un arbore în care orice nod care nu este frunză are ca valoare un operator și are exact doi fii, iar frunzele sunt operanzi. Cu cât prioritatea unui operator este mai mare, cu atât nivelul pe care se află este mai mare. Parantezele influențează prioritatea operatorilor, dar nu apar în arborele asociat.

Indicați expresia corespunzătoare arborelui de mai jos.



- a) $a+b*c+d*e$ b) $a+b*(c+d*e)$ c) $(a+b)*(c+d*e)$
d) $(a+b)*c+d*e$ e) $a+b*(c+d)*e$ f) $(a+b)*c+d*e$

Example:	
$a*(b+c)$	
$a*b+c$	

11. Se consideră un tablou bidimensional **A** cu **n** linii și **n** coloane (**n** – număr natural, **n**>1). Folosind rezolvarea optimă pentru fiecare caz, precizați care dintre problemele următoare se poate rezolva printr-un algoritm de complexitate minimă.

- a) Determinarea numărului de valori nule din **A**
b) Determinarea sumei elementelor de pe diagonala principală
c) Determinarea rangului matricei **A**
d) Ordonarea crescătoare a elementelor de pe prima linie a tabloului prin apelarea celei mai eficiente metode de sortare
e) Determinarea numărului de valori aflate sub diagonala principală
f) Interschimbarea a două coloane ale tabloului

12. Se consideră un graf orientat tare conex cu **n** noduri, numerotate **1, 2, 3, ..., n**. Pentru determinarea drumurilor de lungime minimă de la nodul **1** la celelalte noduri, s-a construit vectorul **t** în care $t[i]=k$ dacă (k,i) este ultimul arc al drumului minim de la nodul **1** la nodul **i**. Precizați instrucțiunea care poate înlocui punctele de suspensie astfel încât apelul **drum(t,n)** să determine afișarea drumului de lungime minimă de la nodul **1** la nodul **n**.

Limbajul C++/ Limbajul C

```
void drum(int t[ ], int i)
{
    if (i!=1) .....
    cout<<i<<" ";
}
```

Limbajul Pascal

```
type vector=array[1..100] of integer;
procedure drum (t:vector; i:integer);
begin
    if i<>1 then .....
    write(i, ' ')
end;
```

- a) `drum(t,n);` b) `drum(t[i],i);` c) `drum(t,t[i]);`
d) `drum(t,i);` e) `drum(t,1);` f) `drum(t[i],t);`

- 13.

Vârful	1	2	3	4
Grad exterior	2	0	2	x
Grad interior	0	2	y	1

Precizați care din următoarele arce aparține grafului orientat cu **4** vârfuri, având gradele din tabelul alăturat ($x, y \in \mathbf{N}$).

- a) (1,2) b) (2,1) c) (2,3) d) (2,4) e) (3,1) f) (4,1)

14. Precizați care este numărul ciclurilor hamiltoniene disticte într-un graf complet cu **5** noduri. (Două cicluri sunt distincte dacă diferă prin cel puțin o muchie.)

- a) 5 b) $4!/2$ c) $4!$ d) $4*4!$ e) $5!$ f) 5^4

15. Precizați câte dintre afirmațiile următoare referitoare la grafuri neorientate sunt adevărate.

1. Dacă gradul oricărui nod este un număr impar, atunci graful trebuie să aibă număr par de noduri.
2. Un ciclu elementar este un caz particular de lanț elementar.
3. Numărul muchiilor grafului nu poate fi mai mic decât numărul nodurilor.
4. Lungimea unui lanț poate fi mai mare decât numărul de noduri al grafului.
5. Numărul valorilor **1** din matricea de adiacență asociată este egal cu dublul numărului de muchii.

- a) 0 b) 1 c) 2 d) 3 e) 4 f) 5

Varianta 4

1. Variabila **n** reprezintă un număr natural cu exact două cifre. Precizați câte dintre expresiile următoare au valoarea **1/true** dacă și numai dacă cifrele lui **n** au aceeași paritate.

Limbajul C++/ Limbajul C

1. $(n/10 - n\%10) \% 2 == 0$
2. $n/10 \% 2 == n \% 2$
3. $n/10 == n \% 10$
4. $(n/10 + n \% 10 * 10) \% 2 == n \% 2$
5. $n/2 == n \% 2$

Limbajul Pascal

1. $(n \text{ div } 10 - n \text{ mod } 10) \text{ mod } 2 = 0$
2. $n \text{ div } 10 \text{ mod } 2 = n \text{ mod } 2$
3. $n \text{ div } 10 = n \text{ mod } 10$
4. $(n \text{ div } 10 + n \text{ mod } 10 * 10) \text{ mod } 2 = n \text{ mod } 2$
5. $n \text{ div } 2 = n \text{ mod } 2$

- a) 0 b) 1 c) 2 d) 3 e) 4 f) 5

2. Pentru implementarea ecuației unei drepte de forma **ax+by+c=0** (unde **a, b, c ∈ ℝ**), se definește structura:

Limbajul C++/ Limbajul C

```
typedef struct
    {float a, b, c;}dreapta;
```

Limbajul Pascal

```
type dreapta=record
    a, b, c : real;
end;
```

Dacă **d1** și **d2** sunt două variabile de tipul dreapta, precizați care dintre următoarele expresii verifică dacă **d1** și **d2** sunt paralele.

Limbajul C++/ Limbajul C

- a) `d1 == d2`
- b) `d1.a==d2.a && d1.b==d2.b`
- c) `a.d1/a.d2==b.d1/b.d2`
- d) `d1.a/d2.a==d1.b/d2.b`
- e) `d1.a==0 && d2.a==0`
- f) `d1.a*d2.b-d1.b*d2.a==0`

Limbajul Pascal

- a) `d1 = d2`
- b) `(d1.a=d2.a) and (d1.b = d2.b)`
- c) `a.d1/a.d2 = b.d1/b.d2`
- d) `d1.a/d2.a=d1.b/d2.b`
- e) `(d1.a=0) and (d2.a=0)`
- f) `d1.a*d2.b-d1.b*d2.a=0`

3. Funcția **f** primește ca parametri două valori reale și returnează cea mai mare dintre cele două valori. Antetul funcției este:

(Limbajul C++/C) `float f(float x, float y);`

(Limbajul Pascal) `function f( x, y: real):real;`

Precizați care dintre următoarele expresii reprezintă suma celor mai mici două valori dintre numerele reale **a, b** și **c**.

- | | |
|---------------------------------------|--|
| a) <code>a+b+c-f(a,b)</code> | b) <code>a+b+c-f(a,b)-f(b,c)</code> |
| c) <code>a+2*b+c-f(a,b)-f(b,c)</code> | d) <code>a+b+c-f(a,b,c)</code> |
| e) <code>a+b+c-f(a,f(c,b))</code> | f) <code>a+b+c-f(f(a,b),f(b,a))</code> |

4. Precizați care sunt valorile afișate în urma execuției următorului program.

Limbajul C++

```
#include<iostream>
using namespace std;
int a,b;
void f(int a, int &b)
{
    if(a>0)
    {
        a++; b--; f(b,a);
    }
}
```

Limbajul C

```
#include<stdio.h>
int a,b;
void f(int a, int *b)
{
    if(a>0)
    {
        a++; (*b)--;
        f(*b,&a);
    }
}
```

Limbajul Pascal

```
var a,b:integer;
procedure f(a:integer;
var b:integer);
begin
    if a>0 then
        begin
            a:=a+1; b:=b-1;
            f(b,a)
        end;
end;
```


<pre> cout<<a<<" "<<b<<" "; } int main() { a=0; b=1; f(b,a); cout<<a<<" "<<b; } </pre>	<pre> printf("%d %d",a,*b); } void main () { a=0; b=1; f(b,&a); printf("%d %d",a,b); } </pre>	<pre> write(a,' ',b,' ') end; begin a:=0; b:=1; f(b,a); write(a,' ',b,' ') end. </pre>
--	--	--

a) -1 2 2 -1 -1 1 b) 0 1 0 1 c) Ciclare infinită
d) -1 2 2 -1 0 1 e) 0 2 0 -1 0 1 f) -1 0 1 -1 1 1

5 În secvența următoare variabilele **n** și **m** au ca valori numere naturale.

· Limbajul C++/ Limbajul C <pre> n:=42015; m:=0; while (n>0) { m=m*100+n/10%10*10+n%10; n/=100; } </pre>	Limbajul Pascal <pre> n:=42015; m:=0; while n>0 do begin m:=m*100+n div 10 mod 10*10+n mod 10; n:=n div 100; end; </pre>
--	--

După rularea secvenței, valoarea variabilei **m** este:

a) 15024 b) 15204 c) 24051 d) 51024 e) 152004 f) 152400

6 Se consideră următorul program:

· Limbajul C++ <pre> #include<iostream> using namespace std; int main () { int n, cn, x=0,p=1; cin>>n; cn=n; while(n) { if (n%10>x)x=n%10; n/=10; } x++; while(cn) { n=n+cn%10*p; p*=x; cn/=10; } cout<<n; } </pre>	Limbajul C <pre> #include<stdio.h> void main () { int n, cn, x=0,p=1; scanf("%d", &n); cn=n; while(n) { if (n%10>x)x=n%10; n/=10; } x++; while(cn) { n=n+cn%10*p; p*=x; cn/=10; } printf("%d", n); } </pre>	Limbajul Pascal <pre> var n,cn,x,p : longint; begin readln(n); cn:=n; x:=0; p:=1; while n>0 do begin if n mod 10>x then x:=n mod 10; n:=n div 10 end; x:=x+1; while cn>0 do begin n:=n+cn mod 10*p; p:=p*x; cn:=cn div 10 end; write(n) end. </pre>
--	--	---

Precizați care este cel mai mic număr natural format din 5 cifre distincte care poate fi citit ca dată de intrare astfel încât valoarea afișată să fie aceeași.

a) 10000 b) 10192 c) 10234 d) 10239 e) 10923 f) 12345

- 7 Tabloul bidimensional **b** (cu liniile și coloanele numerotate de la 1 la n) se obține din tabloul bidimensional **a** prin rotire cu 90° spre dreapta.

De exemplu, dacă **a** este: (1 2 3 4 5 6 7 8 9) se obține tabloul bidimensional **b**:

(7 4 1 8 5 2 9 6 3)

Pentru obținerea unei transformări corecte, secvența:

Limbajul C++/ Limbajul C

```
for(i=1; i<=n; i++)
    for(j=1; j<=n; j++) .....
```

trebuie completată cu atribuirea:

Limbajul C++/ Limbajul C

- a) `b[i][j]=a[j][i]`
- b) `b[i][j]=a[j][n-i+1]`
- c) `b[i][j]=a[n-j+1][n-i+1]`
- d) `b[i][j]=a[n-i+1][n-j+1]`
- e) `b[i][j]=a[n-j+1][i]`
- f) `b[i][j]=a[n-i+1][j]`

Limbajul Pascal

```
for i:=1 to n do
    for j:=1 to n do .....
```

Limbajul Pascal

- a) `b[i][j]:=a[j][i]`
- b) `b[i,j]:=a[j,n-i+1]`
- c) `b[i,j]:=a[n-j+1,n-i+1]`
- d) `b[i,j]:=a[n-i+1,n-j+1]`
- e) `b[i,j]:=a[n-j+1,i]`
- f) `b[i,j]:=a[n-i+1,j]`

- 8 Variabila **x** este de tip întreg și reprezintă o cifră nenulă. Precizați care dintre expresiile următoare este echivalentă cu expresia:

(Limbajul C++/C)

`x == 7 || x == 5`

(Limbajul Pascal)

`(x=7) or (x=5)`

Limbajul C++/ Limbajul C

- a) `35%x==0`
- b) `x!=7&&x!=5`
- c) `x>4&&!(x%2==0||x%3==0)`
- d) `x%2!=0&&x%3!=0`
- e) `!(x!=7||x!=5)`
- f) `x>4&&!(x%2==0&&x%3==0)`

Limbajul Pascal

- a) `35 mod x = 0`
- b) `(x<>7) and (x<>5)`
- c) `(x>4) and not ((x mod 2=0) or (x mod 3=0))`
- d) `(x mod 2<>0) and (x mod 3<>0)`
- e) `not ((x<>7) or (x<>5))`
- f) `(x>4) and not ((x mod 2=0) and (x mod 3=0))`

- 9 Tabloul unidimensional **a** conține **n** numere naturale, ordonate crescător. Se cere afișarea mesajului **DA** dacă în **a** există două elemente a căror diferență este egală cu **s** (număr natural) sau a mesajului **NU**, în caz contrar. Precizați condiția ce trebuie utilizată în locul punctelor de suspensie astfel încât secvența următoare să rezolve corect problema dată.

Limbajul C++/C

```
i = 1; j = 2;
while ( ..... )
{
    if (a[j]-a[i]<s) j++;
    else i++;
}
if (j <= n) cout<<"DA"; |
printf("DA");
else cout<<"NU"; | printf("DA");
```

- a) `j<n`
- b) `j<=n&&a[j]-a[i]!=s`
- c) `j<=n&&a[j]-a[i]==s`
- d) `a[j]-a[i]!=s`
- e) `i<=j`
- f) `i<=n&&a[j]-a[i]==s`

Limbajul Pascal

```
i:= 1; j:= 2;
while ..... do
begin
    if a[j]-a[i]<s then inc(j)
    else inc(i);
end;
if j <=n then write('DA')
else write('NU');
```

- a) `j<n`
- b) `(j<=n) and (a[j]-a[i]<>s)`
- c) `(j<=n) and (a[j]-a[i]=s)`
- d) `a[j]-a[i]<>s`
- e) `i<=j`
- f) `(i<=n) and (a[j]-a[i]=s)`

10. Precizați care este rolul următorului subprogram.

Limbajul C++/ Limbajul C

```
void f(char s[],char t[],int k)
{
    char aux[255];
    strcpy(aux,s+k);
    s[k]=0;
    strcat(s,t);
    strcat(s,aux);
}
```

Limbajul Pascal

```
procedure f(var s:string;t:string;
k:byte);
var aux:string;
begin
    aux:=copy(s,k,255);
    delete(s,k,255);
    s:=concat(s,t);
    s:=concat(s,aux);
end;
```

- a) Șterge ultimele **k** caractere ale lui **s** și concatenează rezultatul cu șirul **t**
 b) Concatenează șirul **s** cu rezultatul concatenării șirurilor **s** și **t**
 c) Inserează șirul **s** în șirul **t**, începând cu poziția **k**
 d) Concatenează șirurile **s** și **t**, obținând un șir de lungime **k**
 e) Înlocuiește primele **k** caractere din **s** cu primele **k** caractere din **t**
 f) Inserează șirul **t** în șirul **s**, începând cu poziția **k**
11. Se consideră un graf orientat cu **6** noduri, numerotate **1,2,...,6**. Arcele grafului sunt de forma **(x,2*x)** pentru orice **x ∈ {1,2,3}** și de forma **(x,x-1)** pentru orice **x ∈ {2,3,4,5,6}**. Care este numărul minim de arce ce trebuie adăugate astfel încât graful să fie tare conex?
- a) 0 b) 1 c) 2 d) 3 e) 4 f) 5
12. Precizați care dintre tablourile următoare poate reprezenta vectorul gradelor unui graf neorientat conex.
- a) (3,2,1,5,1,1) b) (5,1,6,4,5,3) c) (1,1,1,1,2,2)
 d) (1,1,1,1,1,6) e) (2,1,3,1,0,1) f) (1,3,5,2,1,2)
13. Dacă un graf neorientat conex are **n** vârfuri și **3n+2** muchii, precizați care este valoarea minimă pentru **n**.
- a) 16 b) 8 c) 4 d) 2 e) 1 f) 0
14. Pentru un număr natural nenul **n**, se construiește un arbore cu rădăcină astfel: rădăcina este numerotată **n** și orice nod care este numerotat cu o valoare **x > 1** are ca fii nodurile numerotate cu divizorii săi, mai puțin numărul însuși. Toate frunzele arborelui sunt numerotate cu **1**. Precizați câte dintre numerele naturale din intervalul **[10,20]** pot fi alese ca rădăcină, astfel încât arborele asociat să aibă un număr maxim de frunze.
- a) 1 b) 2 c) 3 d) 4 e) 5 f) 6
15. Un pulover norvegian este frumos dacă pentru a-l tricota se folosesc cel puțin **2** și cel mult **4** culori de lână. Precizați câte modalități de combinare a culorilor există pentru a tricota un pulover norvegian frumos, având la dispoziție **5** ghemuri de lână de culori diferite.
- a) 5 b) 12 c) 24 d) 25 e) 48 f) 125

Varianta 5

1. Precizați care dintre următoarele expresii are valoarea **1/true** dacă și numai dacă numărul natural nenul memorat în variabila **x** **nu** este divizibil cu 6.

Limbajul C++/ Limbajul C

- a) $x/6==0$
- b) $x==6$
- c) $x\%6 == 0$
- d) $x/6>0$
- e) $x\%2+x\%3>0$
- f) $x>6$

Limbajul Pascal

- a) $x \text{ div } 6 = 0$
- b) $x=6$
- c) $x \bmod 6 = 0$
- d) $x/6>0$
- e) $x \bmod 2+x \bmod 3>0$
- f) $x>6$

2. Precizați care dintre următoarele instrucțiuni este corectă dacă variabilele **x**, **y** și **z** au declarările de mai jos:

Limbajul C++/ Limbajul C

- ```
float x;
int y,z;
a) x = x*y*z;
b) x = z*y*x;
c) x = x*y*z;
d) x = x*z*x;
e) x = x*z;
f) y = z*x;
```

Limbajul Pascal

- ```
x : real;
y,z:integer;
a) x:= x*y mod z;
b) x:= z mod y*x;
c) x:= x mod y*z;
d) x:= x*z mod x;
e) x:=x mod z;
f) y:=z mod x;
```

3. Precizați ce valoare se va afișa pe ecran în urma executării secvenței de program următoare, știind că **s** este o variabilă care memorează un șir de caractere, iar **i** este o variabilă de tip întreg.

Limbajul C++/ Limbajul C

```
strcpy(s,"admitere");
for(i=0;i<strlen(s);i++)
    if(strchr("politehnica",s[i]))
        strcpy(s+i,s+i+1);
cout<<s; | printf("%s",s);
```

Limbajul Pascal

```
s:='admitere';
for i:=1 to length(s) do
    if pos(s[i],'politehnica')>0 then
        delete(s,i,1);
write(s);
```

- a) **dmt** b) **dm** c) **dmtr** d) **dmr** e) **mt** f) **mrt**

4. Precizați care dintre următoarele afirmații este adevărată pentru **orice** graf neorientat **G** format din **100** de noduri și **100** de muchii.

- a) Graful G nu este conex
- b) Graful G este conex
- c) Graful G este complet
- d) Graful G conține cel puțin un ciclu
- e) Graful G nu are noduri izolate
- f) Graful G conține un lanț elementar de lungime 100

5. Pentru reprezentarea unui graf orientat **G** se utilizează matricea de adiacență. Precizați care este suma elementelor din această matrice dacă graful are **20** de noduri și **30** de arce.
a) **60** b) **50** c) **40** d) **30** e) **20** f) **10**
6. Precizați care este lungimea maximă a unui lanț simplu (lanț în care fiecare muchie apare o singură dată) într-un arbore cu **10** noduri în care fiecare nod are gradul un număr impar.
a) **9** b) **8** c) **7** d) **6** e) **5** f) **4**
7. Tabloul unidimensional **v** conține **n** numere întregi numerotate de la **1** la **n**. Precizați care dintre următoarele secvențe determină înlocuirea primului element din tabloul unidimensional **v** cu cea mai mică valoare care apare în acesta.

Limbajul C++/ Limbajul C

- a) `for(i=1; i<n; i++)
 if (v[i]>v[i+1])
 {
 a=v[i];
 v[i]=v[i+1];
 v[i+1]=a;
 }`
- b) `for(i=n-1; i>=1; i--)
 if (v[i]>v[i+1])
 {
 a=v[i];
 v[i]=v[i+1];
 v[i+1]=a;
 }`
- c) `for(i=n-1; i>=1; i--)
 if (v[i]<v[i+1])
 {
 a=v[i];
 v[i]=v[i+1];
 v[i+1]=a;
 }`
- d) `for(i=1; i<=n-1; i++)
 if (v[i]<v[i+1])
 {
 a=v[i];
 v[i]=v[i+1];
 v[i+1]=a;
 }`
- e) `for(i=n-1; i>=1; i--)
 if (v[i]<v[i+1])
 {
 a=v[i+1];
 v[i]=v[i+1];
 v[i+1]=a;
 }`
- f) `for(i=1; i<=n-1; i++)
 if (v[i]<v[i+1])
 {
 a=v[i+1];`

Limbajul Pascal

- a) `for i:=1 to n-1 do
 if v[i]>v[i+1] then
 begin
 a:=v[i];
 v[i]:=v[i+1];
 v[i+1]:=a;
 end;`
- b) `for i:=n-1 downto 1 do
 if v[i]>v[i+1] then
 begin
 a:=v[i];
 v[i]:=v[i+1];
 v[i+1]:=a;
 end;`
- c) `for i:=n-1 downto 1 do
 if v[i]<v[i+1] then
 begin
 a:=v[i];
 v[i]:=v[i+1];
 v[i+1]:=a;
 end;`
- d) `for i:=1 to n-1 do
 if v[i]<v[i+1] then
 begin
 a:=v[i];
 v[i]:=v[i+1];
 v[i+1]:=a;
 end;`
- e) `for i:=n-1 downto 1 do
 if v[i]<v[i+1] then
 begin
 a:=v[i+1];
 v[i]:=v[i+1];
 v[i+1]:=a;
 end;`
- f) `for i:=1 to n-1 do
 if v[i]<v[i+1] then
 begin
 a:=v[i+1];`

```

        v[i]=v[i+1];
        v[i+1]=a;
    }

```

```

        v[i]:=v[i+1];
        v[i+1]:=a;
    end;

```

8. Precizați pentru care dintre următoarele tablouri unidimensionale se poate aplica algoritmul căutării binare cu scopul de a găsi în mod eficient, dacă există, numere care au cifra unităților egală cu o valoare x , dată.

a) (1, 21, 13, 23, 33, 17, 27) b) (1, 13, 17, 21, 23, 27, 33)
 c) (1, 13, 33, 17, 21, 23, 27) d) (33, 27, 23, 21, 17, 13, 1)
 e) (1, 13, 33, 21, 23, 27, 17) f) (33, 27, 23, 21, 13, 1, 17)

9. În secvența de mai jos, variabila **a** memorează un tablou bidimensional cu **4** linii și **4** coloane, numerotate de la **1** la **4**, cu elementele întregi. Variabila **s** este întreagă, iar **i** este de tip întreg. Precizați care dintre instrucțiunile de mai jos poate înlocui punctele de suspensie, astfel încât secvența să determine memorarea în variabila **s**, a valorii sumei elementelor aflate pe prima și ultima coloană ale matricei.

Limbajul C++/ Limbajul C

```

s=0;
for(i=1;i<=4;i++) ....
a) s=s+a[4][i]+a[i][4];
b) s=s+a[4-i][4]+a[i][1];
c) s=s+a[i][1]+a[i][4];
d) s=s+a[i][i]+a[1][i];
e) s=s+a[1][i]+a[4][i];
f) s=s+a[i][i]+a[5-i][i];

```

Limbajul Pascal

```

s:=0;
for i:=1 to 4 do ....
a) s:=s+a[4,i]+a[i,4];
b) s:=s+a[4-i,4]+a[i,1];
c) s:=s+a[i,1]+a[i,4];
d) s:=s+a[i,i]+a[1,i];
e) s:=s+a[1,i]+a[4,i];
f) s:=s+a[i,i]+a[5-i,i];

```

10. Utilizând metoda backtracking se generează toate anagramele cuvântului **avion**. Precizați câte anagrama încep și se termină cu câte o consoană.

a) 6 b) 12 c) 20 d) 36 e) 38 f) 40

11. Subprogramul **f** are definiția următoare. Dacă variabilele **a** și **b** sunt de tip întreg și memorează valorile **3** respectiv **5**, precizați care vor fi valorile pe care le memorează variabilele **a** și **b** după apelul:

f(a,b); (Limbajul Pascal/C++)

f(a,&b); (Limbajul C).

Limbajul C++

```

void f(int x,int &y)
{int aux;
  aux=x; x=y;
  y=aux;
}

```

Limbajul C

```

void f(int x,int *y)
{int aux;
  aux=x; x=*y;
  *y=aux;
}

```

Limbajul Pascal

```

procedure f(x:integer;var y:integer);
  var aux:integer;
begin
  aux:=x; x:=y; y:=aux;
end;

```

- a) 3 și 3 b) 4 și 3 c) 5 și 5 d) 3 și 5 e) 3 și 4 f) 5 și 3

12. Considerăm declararea următoare, folosită pentru a memora numărătorul și numitorul unei fracții. Precizați care dintre instrucțiunile de mai jos este corectă.

Limbajul C++/ Limbajul C

```
typedef struct
{ int a, b; }fracție;
fracție m,n;
```

- a) `m=n;`
b) `if (m>n) m++;`
c) `if (m==n) m--;`
d) `if (m<=n) m=n;`
e) `if (m!=n) m--;`
f) `if (m>n) m=n;`

Limbajul Pascal

```
type fracție=record
a,b:integer;
end;
var m,n:fracție;
```

- a) `m:=n;`
b) `if (m>n) then m:=m+1;`
c) `if (m=n) then m:=m-1;`
d) `if (m<=n) then m:=n;`
e) `if (m<>n) then m:=m-1;`
f) `if (m>n) then m:=n;`

13. Precizați care este numărul de grafuri orientate distincte formate din 3 noduri și 4 arce. Două grafuri sunt distincte dacă au matricea de adiacență diferită.

- a) 32 b) 30 c) 20 d) 16 e) 15 f) 9

14. Precizați care este instrucțiunea prin care variabilei **y** i se atribuie numărul obținut prin inversarea ordinii cifrelor numărului natural format din exact 2 cifre, memorat în variabila întreagă **x**.

Limbajul C++/ Limbajul C

- a) `y=x/10*10+x%100;`
b) `y=x%10+x/10;`
c) `y=x/10*10+x%10;`
d) `y=x%100/10;`
e) `y=x*10%100+x/10;`
f) `y=x%10/10;`

Limbajul Pascal

- a) `y:=x div 10*10+x mod 100;`
b) `y:=x mod 10 +x div 10;`
c) `y:=x div 10*10+x mod 10;`
d) `y:=x mod 100 div 10;`
e) `y:=x*10 mod 100+x div 10;`
f) `y:=x mod 10 div 10;`

15. Fie **G** un graf neorientat complet cu 100 de noduri. Precizați care dintre următoarele afirmații este adevărată:

- a) În graful **G** există un lanț elementar de lungime 100
b) Graful **G** este un graf hamiltonian
c) Graful **G** este un graf eulerian
d) Graful **G** nu este conex
e) Graful **G** are 900 de muchii
f) Graful **G** are două componente conexe

Varianta 6

1. Precizați care este valoarea maximă pe care o poate avea expresia de mai jos în care **x** este o variabilă de tip întreg.

Limbajul C++/ Limbajul C
 $2 * x \% 10 * 2 \% 10$

Limbajul Pascal
 $2 * x \bmod 10 * 2 \bmod 10$

- a) 9 b) 8 c) 7 d) 0 e) 10 f) 20

2. Variabilele întregi **a** și **b** memorează câte un număr natural nenul. Precizați care dintre următoarele expresii are valoarea **true/1** dacă și numai dacă valorile memorate de **a** și **b** au aceeași paritate.

Limbajul C++/ Limbajul C

Limbajul Pascal

- a) $a == b$
 b) $a \% 2 == 0 \ \&\& \ b \% 2 == 0$
 c) $(a + b) \% 2 == 0$
 d) $a * b \% 2 == 0$
 e) $a \% b == 2$
 f) $a / b == 2$

- a) $a = b$
 b) $(a \bmod 2 = 0) \text{ and } (b \bmod 2 = 0)$
 c) $(a + b) \bmod 2 = 0$
 d) $a * b \bmod 2 = 0$
 e) $a \bmod b = 2$
 f) $a \operatorname{div} b = 2$

3. Precizați ce se afișează în urma executării secvenței de program de mai jos, dacă variabilele **a** și **b** pot memora câte un șir de cel mult 100 de caractere.

Limbajul C++/ Limbajul C

```
strcpy(a, "matematica");
strcpy(b, strstr(a, "ema") + 2);
strcat(b, strchr(a, a[3]) + 1);
cout << b;           | printf("%s", b);
```

Limbajul Pascal

```
a := 'matematica';
b := copy(a, pos('ema', a) + 2, length(a) - pos('ema', a) + 1);
b := b + copy(a, pos(a[4], a) + 1, length(a) - pos(a[4], a) + 1);
write(b);
```

- a) aticamatica b) maticamatica c) maticaaatica
 d) matica e) matematica f) atica

4. Fie un graf neorientat complet cu 10 noduri. Precizați care este numărul minim de muchii care trebuie eliminate astfel încât graful parțial obținut să **nu** fie conex.

- a) 10 b) 9 c) 8 d) 7 e) 6 f) 5

5. Precizați care dintre următoarele șiruri de grade corespund unui graf neorientat cu 6 noduri.

- a) (1, 2, 3, 4, 5, 6) b) (0, 1, 2, 3, 4, 5) c) (0, 1, 0, 1, 0, 1)
 d) (1, 2, 2, 1, 2, 2) e) (1, 1, 1, 1, 1, 2) f) (1, 2, 2, 1, 1, 2)

6. Precizați care este numărul maxim de frunze ce apar într-un arbore cu 17 de noduri, dacă fiecare nod are gradul mai mic sau egal cu 4.

- a) 11 b) 12 c) 13 d) 14 e) 15 f) 16

7. Graful orientat G are 10 noduri și 12 arce. Precizați care este cel mai mare grad exterior al unui nod din acest graf, dacă G este tare conex.

- a) 12 b) 3 c) 10 d) 4 e) 5 f) 6

8. Se consideră un tablou bidimensional a cu n linii și n coloane, numerotate de la 1 la n , cu elemente numere întregi. Precizați ce reprezintă valoarea variabilei întregi x , după executarea secvenței de program de mai jos.

Limbajul C++/ Limbajul C

```
x=0;
for(i=1;i<=n;i++) x=x+a[i][n-i+1];
```

Limbajul Pascal

```
x:=0;
for i:=1 to n do x:=x+a[i,n-i+1];
```

- a) suma elementelor de pe diagonala principală a tabloului a
b) suma elementelor de pe diagonala secundară a tabloului a
c) suma elementelor tabloului a
d) suma elementelor situate pe ultima coloană a tabloului a
e) suma elementelor situate pe prima coloană a tabloului a
f) suma elementelor situate pe ultima linie a tabloului a

9. Se consideră secvența alăturată în care A este un tablou bidimensional cu cinci linii și cinci coloane, numerotate de la 1 la 5, iar x și i sunt variabile de tip întreg. Știind că orice element al tabloului este inițial egal cu numărul de ordine al liniei pe care se află, precizați care este valoarea variabilei x după executarea secvenței de mai jos?

Limbajul C++/ Limbajul C

```
x=0;
for(i=1;i<=5;i++)
if(i%2==0) x=x+A[i-1][i];
```

Limbajul Pascal

```
x:=0;
for i:=1 to 5 do
if ( i mod 2 = 0 ) then x:=x+A[i-1,i];
```

- a) 25 b) 4 c) 15 d) 5 e) 30 f) 3

10. Problema generării tuturor numerelor formate din exact trei cifre nenule, cu toate cifrele distincte două câte două, este similară cu generarea:

- a) aranjamentelor
b) permutărilor
c) elementelor produsului cartezian
d) tuturor submultimilor unei mulțimi
e) combinațiilor
f) partițiilor unei mulțimi

11. O delegație formată din patru elevi ai unei grupe trebuie să participe la o conferință. Știind că în grupă sunt 9 elevi, dintre care cinci sunt fete, precizați care este numărul posibilităților de a forma delegația care va participa la conferință, dacă aceasta trebuie să fie alcătuită din doi băieți și două fete.

- a) 20 b) 45 c) 180 d) 60 e) 90 f) 120

- 12 Fie un șir format din 10000 de numere naturale, fiecare având cel mult 9 cifre. Precizați care dintre următorii algoritmi efectuează un număr minim de pași.
- ordonarea crescătoare a elementelor din șir
 - numărarea elementelor prime din șir
 - determinarea elementului maxim din șir
 - verificarea unicității tuturor elementelor din șir
 - generarea tuturor permutărilor elementelor din șir
 - suma elementelor care apar de exact două ori în șir
- 13 În declarația alăturată, câmpurile **x** și **y** ale înregistrării pot memora coordonatele carteziene ale unui punct din planul xOy. Precizați care dintre următoarele expresii are valoarea **true/1** dacă și numai dacă punctul **p** este situat în cadranul I sau III, dar nu și pe axe.

Limbajul C++/ Limbajul C

```
typedef struct
{
    float x,y;
} punct;
punct p;
```

a) $p.x \cdot p.y > 0$

d) $x.p \cdot y.p > 0$

Limbajul Pascal

```
type punct=record
    x,y:real;
end;
var p:punct;
```

b) $p.x > p.y$

e) $x.p \cdot y.p > 0$

c) $p.x \cdot p.y \geq 0$

f) $p.x + p.y \geq 0$

- 14 Precizați care este suma maximă a elementelor care apar într-un tablou unidimensional cu legături „de tip tată”, asociat unui arbore cu rădăcină format din 10 noduri, etichetate cu numere de la 1 la 10.
- 100
 - 90
 - 81
 - 45
 - 80
 - 60

- 15 Subprogramul **f** are definiția de mai jos. Dacă variabila **a** este de tip întreg și memorează valoarea **3**, precizați care va fi valoarea pe care o memorează aceeași variabilă **a**, după apelul **f(a,a)**; (limbajul Pascal/C++), respectiv **f(&a,&a)**; (în limbajul C).

Limbajul C++

```
void f(int &x,int &y)
{
    x=1;
    x=x+y;
}
```

Limbajul Pascal

```
procedure f(var x,y: integer);
begin
    x:=1;
    x:=x+y;
end;
```

Limbajul C

```
void f(int *x, int *y)
{
    *x=1;
    *x=*x+y;
}
```

a) 1

b) 2

c) 3

d) 4

e) 5

f) 6

Varianta 7

1. Indicați care dintre expresiile C++/C/Pascal de mai jos are valoarea **true/1** dacă și numai dacă numărul memorat în variabila întreagă x aparține reuniunii de intervale:

$[-4, -1] \cup [1, 4] \cup [10, \infty)$.

Limbajul C++/ Limbajul C

- a) $x \geq -4 \ \&\& \ x \leq -1 \ \&\& \ x \geq 1 \ \&\& \ x \leq 4 \ \&\& \ x \geq 10$
- b) $!(x < -4 \ || \ x > -1) \ || \ !(x < 1 \ || \ x > 4) \ || \ !(x < 10)$
- c) $x \geq -4 \ || \ x \leq -1 \ || \ x \geq 1 \ || \ x \leq 4 \ || \ x \geq 10$
- d) $!(x < -4 \ \&\& \ x > 4 \ \&\& \ x > -1 \ || \ x < 1 \ \&\& \ x \geq 10)$
- e) $!(x < -4 \ || \ x > -1) \ \&\& \ !(x < 1 \ || \ x > 4) \ || \ !(x < 10)$
- f) $!(x < -4 \ || \ x > -1) \ \&\& \ !(x < 1 \ || \ x > 4) \ \&\& \ !(x < 10)$

Limbajul Pascal

- a) $(x \geq -4) \text{ and } (x \leq -1) \text{ and } (x \geq 1) \text{ and } (x \leq 4) \text{ and } (x \geq 10)$
- b) $\text{not}((x < -4) \text{ or } (x > -1)) \text{ or } \text{not}((x < 1) \text{ or } (x > 4)) \text{ or } \text{not}(x < 10)$
- c) $(x \geq -4) \text{ or } (x \leq -1) \text{ or } (x \geq 1) \text{ or } (x \leq 4) \text{ or } (x \geq 10)$
- d) $\text{not}((x < -4) \text{ and } (x > 4) \text{ and } (x > -1) \text{ or } (x < 1) \text{ and } (x \geq 10))$
- e) $\text{not}((x < -4) \text{ or } (x > -1)) \text{ and } \text{not}((x < 1) \text{ or } (x > 4)) \text{ or } \text{not}(x < 10)$
- f) $\text{not}((x < -4) \text{ or } (x > -1)) \text{ and } \text{not}((x < 1) \text{ or } (x > 4)) \text{ and } \text{not}(x < 10)$

2. Indicați expresia C++/C/Pascal care are valoarea **true /1**:

Limbajul C++/ Limbajul C

- a) $\text{floor}(5)+1 == \text{ceil}(5)$
- b) $\text{floor}(5.49) == \text{ceil}(5.49)$
- c) $\text{floor}(5.19) == \text{floor}(5.91)$
- d) $\text{floor}(5.91) == \text{ceil}(5.19)$
- e) $\text{floor}(\text{sqrt}(8)) == \text{ceil}(\text{sqrt}(8))$
- f) $\text{sqrt}(4) == \text{pow}(4,2)$

Limbajul Pascal

- a) $\text{trunc}(5)+1 = \text{round}(5)$
- b) $\text{trunc}(5.19) = \text{round}(5.91)$
- c) $\text{trunc}(5.19) = \text{trunc}(5.91)$
- d) $\text{round}(5.91) = \text{round}(5.19)$
- e) $\text{round}(\text{sqrt}(8)) = \text{trunc}(\text{sqrt}(8))$
- f) $\text{sqrt}(4) = \text{sqr}(4)$

3. Se consideră două tablouri unidimensionale **A** și **B**. Știind că **A** = (7, 10, 12, 18, 20), iar în urma interclasării tablourilor **A** și **B**, în ordine descrescătoare, se obține tabloul cu elementele (46, 20, 18, 17, 12, 10, 10, 7, 4, 3). Atunci tabloul **B** poate fi:

- a) (3, 4, 17, 46) b) (3, 4, 10, 46) c) (3, 4, 10, 17)
- d) (3, 4, 10, 17, 46) e) (46, 17, 4, 3) f) (46, 10, 4, 3)

4. Pentru a verifica dacă într-un tablou unidimensional având elementele (3, 4, 7, 10, 12, 17, 18, 20, 46) există elementul cu valoarea $x=17$, se aplică metoda căutării binare. Știind că numerotarea elementelor, în tablou, se realizează începând cu poziția 0, care este numărul minim de elemente ale tabloului care trebuie verificate pentru a găsi elementul căutat?

- a) 6 b) 2 c) 9 d) 5 e) 3 f) 1

5. Variabila **x** este de tip real și poate memora un număr real din intervalul **[32, 48]**. Numărul valorilor distincte pe care le poate avea expresia următoare este:

Limbajul C++/C
`floor(sqrt(x+1))`

Limbajul Pascal
`trunc(sqrt(x+1))`

- a) 17 b) 1 c) 0 d) 2 e) 4 f) 3

6. Un grup format din șase prieteni (**Andrei, Bogdan, Claudiu, Daniel, Emil, Florin**) dorește să participe la o competiție de baschet pentru echipe formate din câte trei jucători. Știind că echipa **Andrei, Bogdan, Claudiu** este identică cu echipa **Bogdan, Claudiu, Andrei**, precizați care este numărul de echipe care se pot forma cu cei șase prieteni.

- a) 2 b) 720 c) 120 d) 20 e) 6 f) 3

7. Fie subprogramul recursiv următor:

Limbajul C++
`void ex(char c)
{ if (c>'a')
 ex(c-1);
 cout<< c;
 if (c>'a')
 ex(c-1);
}`

Limbajul C
`void ex(char c)
{ if (c>'a')
 ex(c-1);
 printf("%c", c);
 if (c>'a')
 ex(c-1);
}`

Limbajul Pascal
`procedure ex(c:char)
begin
 if (c>'a') then
 ex(pred(c));
 write(c);
 if (c>'a') then
 ex(pred(c));
end;`

Indicați numărul de autoapeluri ale subprogramului dacă se apelează **ex('c')**:

- a) 0 b) 1 c) 7 d) 3 e) 6 f) 5

8. Într-un program C++/C/Pascal în care **a** este o variabilă de tip întreg, se citesc datele din fișierul "**admitere.dat**" utilizând următoarea instrucțiune:

Limbajul C++
`f>>a;`

Limbajul C
`fscanf(f, "%d", &a);`

Limbajul Pascal
`readln ( f, a);`

Precizați care este forma corectă a instrucțiunii ce are ca efect închiderea fișierului utilizat:

Limbajul C++
a) `close(f);`
b) `close(admitere);`
c) `admitere.close();`
d) `close.admitere;`
e) `close.f;`
f) `f.close();`

Limbajul C
a) `fclose(admitere);`
b) `close(admitere);`
c) `close(f);`
d) `admitere(close);`
e) `close.f`
f) `fclose(f);`

Limbajul Pascal
a) `f.close();`
b) `admitere.close();`
c) `close(admitere);`
d) `admitere(close);`
e) `close.f;`
f) `close(f);`

9. Se consideră următoarea secvență de instrucțiuni:

Limbaajul C++	Limbaajul C	Limbaajul Pascal
<pre>char cif; int cifra; cin>>cif; cifra = cif - '0'; cout<< cifra;</pre>	<pre>char cif; int cifra; scanf("%c", &cif); cifra = cif-'0'; printf ("%d", cifra);</pre>	<pre>var cif: char; cifra: integer; begin read(cif); cifra:=ord(cif)-ord('0'); write (cifra); end.</pre>

Precizați ce se afișează după executarea acestei secvențe dacă, în urma operației de citire, variabila **cif** conține caracterul '9'.

- a) instrucțiunea de atribuire b) '9' c) 9 d) 0 e) '0' f) 57
- `cifra:=ord(cif)-ord('0');`
{Pascal}
`cifra = cif - '0'; //C++/C`
este incorectă

10 Variabila **c** definită mai jos, memorează codul, cele două note obținute la probele matematică și informatică din cadrul concursului de admitere la Facultatea de Automatică și Calculatoare, precum și media obținută la examenul de bacalaureat pentru un candidat.

Limbaajul C++	Limbaajul C	Limbaajul Pascal
<pre>typedef struct { unsigned cod; float p1, p2; float medbac; }candidat ; candidat c;</pre>	<pre>typedef struct { unsigned cod; float p1, p2; float medbac; }candidat; candidat c;</pre>	<pre>type candidat=record cod: word; p1, p2: real; medbac: real; end; var c: candidat;</pre>

Precizați care este expresia corectă ce poate fi utilizată pentru a verifica dacă un candidat îndeplinește baremul minim de admitere (media de admitere este minimum 5):

- a) $(p1+p2)/2*0.8 + medbac*0.2 \geq 5.0$
- b) $(candidat.p1 + candidat.p2)*0.8 + candidat.medbac*0.2 \geq 5.0$
- c) $(candidat.c.p1 + candidat.c.p2)*0.8 + candidat.c.medbac*0.2 \geq 5.0$
- d) $(c.p1 + c.p2)/2*0.8 + c.medbac*0.2 \geq 5.0$
- e) $((candidat.p1 + candidat.p2)*0.8 + candidat.medbac*0.2) \geq 5.0$
- f) $80/100*(c.p1 + c.p2)/2+ 20/100*c.medbac \geq 5.0$

11. Precizați care este antetul corect al unui subprogram (definit de utilizator) care returnează prima și ultima cifră a unui număr natural n , fără a permite modificarea parametrului n .

Limbajul C++

- a) unsigned cifre (unsigned n)
- b) unsigned cifre (unsigned n, unsigned prim, unsigned ult)
- c) void cifre (unsigned n, unsigned prim, unsigned ult)
- d) void cifre (unsigned n, unsigned &prim, unsigned &ult)
- e) void cifre (unsigned &n, unsigned &prim, unsigned &ult)
- f) void cifre (unsigned &n)

Limbajul C

- a) unsigned cifre (unsigned n)
- b) unsigned cifre (unsigned n, unsigned prim, unsigned ult)
- c) void cifre (unsigned n, unsigned prim, unsigned ult)
- d) void cifre (unsigned n, unsigned *prim, unsigned *ult)
- e) void cifre (unsigned *n, unsigned *prim, unsigned *ult)
- f) void cifre (unsigned *n)

Limbajul Pascal

- a) function cifre (n: longint): byte;
- b) function cifre (n: longint; prim,ult: byte): byte;
- c) procedure cifre (n: longint; prim,ult: byte);
- d) procedure cifre (n: longint; var prim,ult: byte);
- e) procedure cifre (var n: longint; var prim,ult: byte);
- f) procedure cifre (var n: longint);

12. Precizați care sunt numărul maxim, respectiv numărul minim de componente conexe pentru un graf neorientat cu **16** noduri și **16** muchii?

- a) 1 și 1
- b) 11 și 2
- c) 2 și 1
- d) 16 și 1
- e) 11 și 1
- f) 10 și 1

13. Precizați care sunt numărul minim și numărul maxim de arce ale unui graf orientat tare conex cu **15** vârfuri.

- a) 14 și 105
- b) 15 și 105
- c) 15 și 210
- d) 14 și 210
- e) 15 și 15
- f) 14 și 15

14. Dacă G este un graf neorientat eulerian cu **10** noduri și **16** muchii, iar lista de adiacență a fiecărui nod din G este formată din cel puțin un element, precizați care dintre afirmațiile de mai jos sunt întotdeauna adevărate.

- 1. G este conex
- 2. G are cel puțin un nod de grad egal cu 2
- 3. G este hamiltonian
- 4. G nu poate conține cicluri elementare de lungime 3.

- a) toate
- b) niciuna
- c) 1, 2, 3
- d) 2, 3
- e) 3, 4
- f) 1, 2

15. Un arbore binar este un arbore cu rădăcină în care orice nod are cel mult doi fii. Înălțimea unui arbore binar este dată de lungimea celui mai lung lanț elementar care are una dintre extremități în rădăcină și cealaltă în oricare dintre frunze. Numărul maxim de noduri dintr-un arbore binar de înălțime 5 este:
- a) 31 b) 15 c) 32 d) 63 e) 64 f) 6

Varianta 8

1. Indicați care dintre expresiile C++/C/Pascal de mai jos are valoarea **true/1**, dacă și numai dacă numărul memorat în variabila întreagă **x** nu aparține reuniunii de intervale:

$[-4, -1] \cup [1, 4] \cup [10, \infty)$.

Limbajul C++/ Limbajul C

- a) `!((x>=-4 && x<=-1)&&(x>=1 && x<=4)&&(x>=10))`
- b) `(x<-4 || x>-1) || (x<1 || x>4) || (x<10)`
- c) `(x<-4 && x>-1) || (x<1 && x>4) || (x<10)`
- d) `!(x<-4 && x>4 && x>-1 || x<1 && x>=10)`
- e) `!((x>=-4 && x<=-1)|| (x>=1 && x<=4)|| (x>=10))`
- f) `!(x>=-4 && x<=-1)||!(x>=1 && x<=4)||!(x>=10)`

Limbajul Pascal

- a) `not((x>=-4 and x<=-1)and(x>=1 and x<=4)and(x>=10))`
- b) `(x<-4 or x>-1) or (x<1 or x>4) or (x<10)`
- c) `(x<-4 and x>-1) or (x<1 and x>4) or (x<10)`
- d) `not(x<-4 and x>4 and x>-1 or x<1 and x>=10)`
- e) `not((x>=-4 and x<=-1)or(x>=1 and x<=4)or(x>=10))`
- f) `not(x>=-4 and x<=-1)or not(x>=1 and x<=4)or not(x>=10)`

2. O expresie C++/C/Pascal care are valoarea **true /1** este:

Limbajul C++/ Limbajul C

- a) `floor(5)+1 == ceil(5)`
- b) `floor(5.49) == ceil(5.49)`
- c) `floor(5.19) == floor(5.91)`
- d) `floor(5.91) == ceil(5.91)`
- e) `floor(sqrt(8))==ceil(sqrt(8))`
- f) `sqrt(4) == pow(4,2)`

Limbajul Pascal

- a) `trunc(5)+1=round(5)`
- b) `trunc(5.19)=round(5.91)`
- c) `trunc(5.19)=trunc(5.91)`
- d) `trunc(5.91)=round(5.91)`
- e) `round(sqrt(8))=trunc(sqrt(8))`
- f) `sqrt(4) = sqr(4)`

3. Pentru a verifica dacă într-un tablou unidimensional există elementul cu valoarea **x=17**, se aplică metoda căutării binare, iar succesiunea de elemente ale tabloului a căror valoare se compară cu valoarea lui **x**, pe parcursul aplicării metodei indicate, este: **12,18,17**. Numerotarea elementelor, în tablou, se realizează începând cu poziția 0. Elementele tabloului pot fi (în ordinea în care apar în tablou):

- a) (3,4,7,12,15,17,18,20)
- b) (3,7,8,10,12,17,18,20,46)
- c) (4,7,12,17,18,20,46)
- d) (3,4,7,10,12,18,17,20)
- e) (3,7,8,10,12,17,18,20)
- f) (3,4,7,10,12,17,18,20,46)

4. Se consideră două tablouri unidimensionale **A** și **B**. Știind că în urma interclasării tablourilor **A** și **B** în ordine descrescătoare se obține tabloul cu elementele: **(46,20,18,17,12,10,10,7,4,3)**, o variantă corectă pentru valorile celor două tablouri este:

- a) (3,4,17,46) și (7,10,12,18,20)
- b) (7,10,12,18) și (46,17,4,3)
- c) (3,4,10,46) și (7,10,12,18,20)
- d) (7,10,12,18,20) și (46,17,4,3)

e) (3,4,10,17,46) și (7,10,12,18,20) f) (3,4,17,46) și (7,10,12,20)

5. Variabila **x** este de tip întreg și poate memora un număr natural format din exact două cifre. Indicați cea mai mare valoare pe care o poate avea expresia :

Limbajul C++/ Limbajul C
abs(x/10-x%10)

Limbajul Pascal
| abs(x div 10- x mod 10)

a) 1 b) 10 c) 9 d) 8 e) 5 f) 0

6. O echipă profesionistă de ciclism este alcătuită din 8 sportivi. La fiecare mare tur participă doar cu un lot format din 4 cicliști. Precizați care este numărul de variante pentru formarea lotului de cicliști ce pot concura la Turul Franței în anul 2020.

a) 40320 b) 0 c) 1680 d) 70 e) 8 f) 4

7. Fie subprogramul recursiv următor în care **n** este un număr natural nenul:

Limbajul C++/ Limbajul C
unsigned ex(unsigned n)
{ unsigned a;
 if (n == 0) return 9;
 else
 { a= ex(n / 10);
 if (n % 10 < a)
 return n%10;
 return a;
 }
}

Limbajul Pascal
function ex(n:longint): byte;
var a: byte;
begin
 if n=0 then ex:=9
 else begin
 a:= ex(n div 10);
 if n mod 10 < a then
 ex:= n mod 10
 else ex:=a;
 end;
end;

Precizați pentru care dintre valorile următoare se va returna un număr impar la apelul funcției **ex(n)**.

a) 90 b) 98 c) 709 d) 340 e) 512 f) 256

8. Într-un program C++/C/Pascal, variabila **a** este de tip întreg, iar datele din fișierul "**candidati.dat**" se citesc utilizând următoarea instrucțiune:

Limbajul C++
f>>a;

Limbajul C
| fscanf(f, "%d", &a);

Limbajul Pascal
| readln (f, a);

Precizați care este forma corectă a instrucțiunii ce are ca efect închiderea fișierului utilizat:

Limbajul C++
a) close(f);
b) close(candidati);
c) candidati.close();
d) f.close();
e) close.f;
f) f(close);

Limbajul C
a) fclose(candidati);
b) close(candidati);
c) close(f);
d) fclose(f);
e) close.f;
f) f(close);

Limbajul Pascal
a) f.close();
b) candidati.close();
c) close(candidati);
d) close(f);
e) f(close);
f) close(candidati);

9. Se consideră următoarea secvență de instrucțiuni:

Limbajul C++

```
char cif; int cifra;
cin>>cifra;
cif = cifra + '0';
cout<<cif;
```

Limbajul C

```
char cif; int cifra;
scanf("%d", &cifra);
cif = cifra + '0';
printf ("%c", cif);
```

Limbajul Pascal

```
var cif: char;
    cifra: integer;
begin
    read(cifra);
    cif:=chr(cifra+ord('0'));
    write (cif);
end.
```

Precizați care este valoarea memorată în variabila **cif** la finalul secvenței dacă, după citire, variabila **cifra** conține valoarea 9.

a) instrucțiunea de atribuire b) '9' c) 9 d) '0' e) 0 f) 57
cif:=chr(cifra+ ord('0'));{Pascal}
cif = cifra + '0'; //C++/C este
 incorectă

10. Considerând declarațiile de mai jos,

Limbajul C++/ Limbajul C

```
typedef struct{
    unsigned z, l;}datan;
typedef struct{
    char nume[30];
    char sex;
    datan dn;
}elev;
elev e;
```

Limbajul Pascal

```
type datan=record
    z, l: byte; end;
elev=record
    nume: string[30];
    sex: char;
    dn: datan;
end;
var e: elev;
```

Precizați care este expresia corectă pentru a verifica dacă elevul este fată și este născută în primele zece zile ale lunii iulie:

Limbajul C++/ Limbajul C

- a) **e.sex=='F' && e.sex=='f' && e.datan.l==7 && e.datan.z<=10**
- b) **(elev.sex=='F' || elev.sex=='f') && (elev.dn.l==7 && elev.dn.z<=10)**
- c) **(e.sex=='F' || e.sex=='f') && (e.dn.l==7 && e.dn.z<10)**
- d) **(e.sex=='F' && e.sex=='f') && (e.dn.l==7 && e.dn.z<=10)**
- e) **e.sex=='F' || e.sex=='f' || e.dn.l==7 && e.dn.z<=10**
- f) **(e.sex=='F' || e.sex=='f') && (e.dn.l==7 && e.dn.z<=10)**

Limbajul Pascal

- a) **(e.sex='F') and (e.sex='f') and (e.datan.l=7) and (e.datan.z<=10)**
- b) **((elev.sex='F') or (elev.sex='f')) and (elev.dn.l=7) and (elev.dn.z<=10)**
- c) **((e.sex='F') or (e.sex='f')) and ((e.dn.l=7) and (e.dn.z<10))**
- d) **((e.sex='F') and (e.sex='f')) and (e.dn.l=7) and (e.dn.z<=10)**
- e) **((e.sex='F') or (e.sex='f')) or (e.dn.l=7) and (e.dn.z<=10)**
- f) **((e.sex='F') or (e.sex='f')) and (e.dn.l=7) and (e.dn.z<=10)**

11. Pentru subprogramul următor:

Limbajul C++/ Limbajul C

```
unsigned suma( unsigned n)
{ unsigned s=0;
  while(n)
  { s+=n%10;
    n/=10;
  }
  return s;
}
```

Limbajul Pascal

```
function suma(n:longint): byte;
var s: byte;
begin
  s:=0;
  while n<>0 do begin
    s:= s+ n mod 10;
    n:= n div 10;
  end;
  suma:=s;
end;
```

Precizați care dintre variante **nu** reprezintă o variantă corectă de apel.

Limbajul C++

- a) if (suma(n) % 2) cout<<"NU";
 else cout<<"DA";
- b) cout<<suma(10945);
- c) cout<<suma(12,12);
- d) cout<<suma(suma(n)+suma(10945));
- e) cout<<suma(n);
- f) sn= suma(n); // sn este o variabilă declarată de tip unsigned

Limbajul C

- a) if (suma(n) % 2) printf("NU");
 else printf("DA");
- b) printf("%d",suma(10945));
- c) printf("%d", suma(12,12));
- d) printf("%d",suma(suma(n)+suma(10945)));
- e) printf("%d",suma(n));
- f) sn= suma(n); // sn este o variabilă declarată de tip unsigned

Limbajul Pascal

- a) if suma(n) mod 2 = 0 then write ('DA')
 else write('NU');
- b) write(suma(10945));
- c) write(suma(12,12));
- d) write(suma(suma(n)+suma(10945)));
- e) write(suma(n));
- f) sn:=suma(n); {sn este o variabilă declarată de tip byte }

12. Precizați care este numărul minim de muchii ale unui graf neorientat cu **16** noduri care are exact două componente conexe, fiecare dintre aceste două componente fiind graf complet.
- a) 15 b) 14 c) 105 d) 55 e) 28 f) 56
13. Se cunosc următoarele informații despre matricea de adiacență a unui graf neorientat conex: are 10 linii și 24 de valori nenule. Precizați care este valoarea maximă pe care o poate avea gradul unui nod într-un astfel de graf.
- a) nu există astfel de graf b) 12 c) 1
d) 10 e) 9 f) 8

14. Fie graful orientat G , definit prin perechea ordonată de mulțimi $X=\{1,2,3,4,5\}$ și $U=\{(1,2), (2,1), (2,3), (3,4), (4,3), (4,1), (4,5), (5,1), (1,5)\}$.
Precizați care dintre afirmațiile următoare nu este adevărată pentru acest graf.

- a) Graful este conex
- b) Graful este tare conex
- c) Graful are două componente conexe
- d) Graful conține cel puțin un drum elementar de lungime 4.
- e) Graful nu conține vârfuri izolate (vârf izolat = vârf care nu este adiacent cu alt vârf)
- f) Graful conține cel puțin un circuit elementar.

15. Precizați care este numărul maxim de frunze ale unui arbore binar (arbore în care fiecare nod are cel mult doi fii) cu 66 de noduri.

- a) 65 b) 35 c) 1 d) 33 e) 32 f) 66

Varianta 9

1. Fie n un număr natural cu cel puțin 4 cifre. Precizați care dintre următoarele instrucțiuni determină interschimbarea cifrei sutelor cu cifra zecilor.

Limbajul C++/ Limbajul C

1. $n = n \% 10 + n / 1000 * 1000 + n \% 1000 / 100 * 10 + n \% 100 / 10 * 100$;
2. $n = n / 1000 * 1000 + n \% 1000 / 100 * 100 + n \% 100 / 10 * 10 + n \% 10$;
3. $n = n \% 1000 / 100 * 10 + n \% 100 / 10 * 100 + n \% 10 + n / 1000 * 1000$;
4. $n = n \% 1000 / 100 * 10 + n \% 100 / 10 * 100 + n / 10$;

Limbajul Pascal

1. $n := n \bmod 10 + n \operatorname{div} 1000 * 1000 + n \bmod 1000 \operatorname{div} 100 * 10 + n \bmod 100 \operatorname{div} 10 * 100$;
 2. $n := n \operatorname{div} 1000 * 1000 + n \bmod 1000 \operatorname{div} 100 * 100 + n \bmod 100 \operatorname{div} 10 * 10 + n \bmod 10$;
 3. $n := n \bmod 1000 \operatorname{div} 100 * 10 + n \bmod 100 \operatorname{div} 10 * 100 + n \bmod 10 + n \operatorname{div} 1000 * 1000$;
 4. $n := n \bmod 1000 \operatorname{div} 100 * 10 + n \bmod 100 \operatorname{div} 10 * 100 + n \operatorname{div} 10$;
- a) 1 și 2 b) 1 și 3 c) 1, 3 și 4
d) 1 și 4 e) 2 și 3 f) 2 și 4

2. Dacă expresia:

Limbajul C++/ Limbajul C
 $!(x > 2) || (x \leq 5) \&\& (x > -5)$

Limbajul Pascal
 $|\text{ not } (x > 2) \text{ or } (x \leq 5) \text{ and } (x > -5)$

este adevărată, atunci:

- a) $x \in (-5, 2) \cup [5, \infty)$ b) $x \in (-\infty, 2] \cap [5, \infty)$ c) $x \in (-\infty, 2] \cup [5, \infty)$
d) $x \in (-\infty, 5)$ e) $x \in (-5, 2) \cap [5, \infty)$ f) $x \in (-\infty, 5]$

3. În urma executării secvenței de program de mai jos, în care variabila s memorează un șir cu cel mult 100 caractere, iar i este de tip întreg, se afișează șirul **acbb**. Precizați care este conținutul șirului s înainte de această secvență.

Limbajul C++/C

```
for(i=0; i<strlen(s); i++)
{
    strcpy(s+i, s+i+1);
    if(!strchr("aeiou", s[i]))
        s[i]--;
    else s[i]++;
}
cout<<s;      |
printf("%s", s);
```

Limbajul Pascal

```
for i:=1 to length(s) do
begin
    delete(s, i, 1);
    if pos(s[i], 'aeiou')=0 then
        s[i]:=chr(ord(s[i])-1)
    else s[i]:=chr(ord(s[i])+1);
end;
write(s);
```

- a) abaa b) abcdbdcb c) abcd
d) abcdcdcb e) acbb f) bdcc

4. Precizați care dintre următoarele matrice, este matricea de adiacență a unui arbore cu 4 noduri.

- a) $\begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ b) $\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ c) $\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$

	1 0 0 0		1 0 0 0		1 1 0 0
	1 0 1 0		0 1 0 0		1 0 0 0
d)	0 0 1 0	e)	0 0 1 0	f)	0 0 1 0
	0 0 0 1		0 1 0 1		0 0 0 1
	1 0 0 1		1 0 0 1		1 0 0 1
	0 1 1 0		0 1 1 0		0 1 1 1

5. În secvența de mai jos, variabilele **i** și **j** sunt de tip întreg, iar variabila **a** memorează un tablou bidimensional în care prima linie și prima coloană sunt numerotate începând cu 1. Toate elementele tabloului primesc valori în urma executării secvenței.

Limbajul C++/ Limbajul C

```
for (j=1; j<=4; j++)
  for (i=3; i>=1; i--)
    if (j==1 || i==3) a[i][j]=i+j-1;
    else
      a[i][j]=a[i][j-1]+a[i+1][j];
```

Limbajul Pascal

```
for j:=1 to 4 do
  for i:=3 downto 1 do
    if (j=1) or (i=3) then
      a[i][j]:=i+j-1
    else
      a[i][j]:=a[i][j-1]+a[i+1][j];
```

Precizați câte numere prime sunt memorate în tabloul **a** după executarea secvenței de program.

- a) 3 b) 4 c) 5
d) 6 e) 7 f) 8
6. Se generează prin metoda backtracking mulțimi distincte cu elemente numere naturale nenule și cu proprietatea că suma elementelor fiecărei mulțimi este egală cu 6 astfel: {1, 2, 3}, {1, 5}, {2, 4}, {6}. Folosind aceeași metodă pentru a genera mulțimi distincte cu elemente numere naturale nenule și cu proprietatea că suma elementelor fiecărei mulțimi este egală cu 10, stabiliți în ce ordine sunt generate următoarele mulțimi:
1) {2, 3, 5}; 2) {3, 7}; 3) {2, 8}; 4) {1, 9}.
a) 4 1 2 3 b) 4 1 3 2 c) 4 2 1 3
d) 4 2 3 1 e) 4 3 1 2 f) 4 3 2 1
7. Se consideră graful **neorientat** $G=(X,U)$ unde $X=\{1,2,3,4,5,6\}$ și $U=\{[1,2], [1,3], [5,1], [3,4], [4,5], [3,2], [3,5]\}$. Precizați câte cicluri elementare distincte există în graful G . (Două cicluri elementare sunt distincte dacă diferă prin cel puțin o muchie).
a) 2 b) 3 c) 4
d) 5 e) 6 f) 7
8. Se consideră graful **orientat** $G=(V, E)$ unde $V=\{1,2,3,4,5,6,7\}$ și $E=\{(1,2), (6,1), (2,5), (2,3), (4,5), (3,4), (3,6)\}$. Precizați câte componente **tare conex**e are graful dat.
a) 1 b) 2 c) 3
d) 4 e) 5 f) 6
9. Un arbore cu nodurile numerotate de la 1 la 12, este memorat cu ajutorul vectorului de tați **tata**= (2,5,5,3,0,2,3,7,6,6,7,4). Numărul de lanțuri elementare de lungime maximă care leagă două noduri ale arborelui este:
a) 2 b) 3 c) 4

d) 5

e) 6

f) 7

10. Se consideră declarațiile următoare, în care variabila **x** memorează coordonatele, în planul **xOy**, ale centrului unui cerc, precum și lungimea razei acestuia.

Limbajul C++/ Limbajul C

```
typedef struct {
    float x, y;
} punct;
typedef struct {
    punct c;
    float r;
} cerc;
cerc x;
```

Limbajul Pascal

```
type punct=record
    x,y:real;
end;
cerc=record
    c:punct;
    r:real;
end;
var x:cerc;
```

Expresia care verifică dacă originea sistemului de coordonate, este în interiorul cercului, este:

- a) $c.x*c.x+c.y*c.y<c.r*c.r$; b) $x.c.x+x.c.y<x.r$
 c) $c.x*c.x+c.y*c.y<x.r*x.r$ d) $x.x*x.x+x.y*x.y<x.r*x.r$
 e) $x.c.x*x.c.x+x.c.y*x.c.y<x.r*x.r$ f) $x.r*x.r<x.c.x*x.c.x+x.c.y*x.c.y$

11. Se consideră tabloul unidimensional **x=(1,2,4,3)**. Precizați care dintre următoarele variante reprezintă tabloul unidimensional **y**, știind că pentru orice $0 \leq i < 4$, există relația **x[y[i]]=y[x[i]]**.

- a) **y=(1,2,3,4)** b) **y=(1,3,4,2)** c) **y=(2,3,1,4)**
 d) **y=(3,2,1,4)** e) **y=(3,4,1,2)** f) **y=(4,2,1,3)**

12. Subprogramul **f** este definit mai jos. Precizați ce valori se vor afișa în urma apelului **f('m',0)**.

Limbajul C++/ Limbajul C

```
void f(char c, int x)
{
    if(!strchr("aeiou",c)){
        f(c-1,x+1)
        cout<<c;|printf("%c",c);
    }else
        cout<<x;|printf("%d",x);
}
```

Limbajul Pascal

```
procedure f(c:char; x: word);
begin
    if (pos(c,'aeiou')=0) then
        begin
            f(pred(c),x+1);
            write(c); end
        else
            write(x);
    end;
```

- a) 4jklm b) 4jmk1 c) 4mlkj
 d) jklm4 e) jmk14 f) mlkj4

13. Se consideră subprogramele de mai jos. Dacă înaintea apelului **g(x)**, variabilele globale de tip întreg **x** și **y** aveau valorile 1, respectiv -3, precizați care vor fi valorile memorate în variabilele globale **x** și **y** după executarea apelului **g(x)**.

Limbajul C++/ Limbajul C

```
void f(int x)
{ x=x+1;
```

Limbajul Pascal

```
procedure f(x: longint);
begin
```

```

    y=2*x+3; }
void g(int x)
{
    int a,b;
    a=x+y;    b=x-y;
    f(a);    f(b);
    y=y+b;
}

```

a) 1 și 13

d) 2 și 13

b) 1 și 17

e) 2 și 17

```

x:=x+1;
y:=2*x+3;end;
procedure g(x: longint);
var a,b:longint;
begin
    a:=x+y;    b:=x-y;
    f(a);    f(b);
    y:=y+b;end;

```

c) 1 și 18

f) 2 și 18

14. În secvența de program de mai jos, toate variabilele sunt de tip întreg. Precizați care este valoarea afișată la finalul executării secvenței următoare.

Limbajul C++/Limbajul C

```

k=0;
for( i=1;i<=9999;i++)
{
    d=2;p=1;n=i;
    while(d*d<=n)
    {
        e=0;
        while(n%d==0)
        {
            e++;
            n=n/d;
        }
        p=p*(e+1);
        d++;
    }
    if(n>1)p=p*2;
    if(p%2==0) k++;
}
cout<<k; | printf("%d",k);

```

a) 99

d) 9900

b) 8901

e) 9901

Limbajul Pascal

```

k:=0;
for i:=1 to 9999 do
begin
    d:=2;p:=1;n:=i;
    while d*d<=n do
    begin
        e:=0;
        while n mod d=0 do
        begin
            inc(e);
            n:=n div d;
        end;
        p:=p*(e+1);
        inc(d);
    end;
    if n>1 then p:=p*2;
    if p mod 2=0 then inc(k);
end;
write(k);

```

c) 8990

f) 9990

15. Funcția f este definită mai jos.

Limbajul C++/ Limbajul C

```

int a[100];
int f(int x[100], int st, int dr)
{
    if(st==dr) return 1;
    else
        if(f(x,st,(st+dr)/2)>0&&f(x,(st+dr)/2+1,dr)>0)
            if (x[(st+dr)/2+1]>x[(st+dr)/2]) return 1;
            else return 0;
        else return 0;
}

```

Limbajul Pascal

```

type vector=array[0..99] of integer;

```



```

var a:vector;
function f(x:vector;st,dr:integer):byte;
begin
if st=dr then f:=1
else
    if (f(x,st,(st+dr)div 2)>0) and (f(x,(st+dr)div 2+1,dr)>0) then
        if x[(st+dr)div 2+1]>x[(st+dr)div 2] then f:=1
        else f:=0
    else f:=0;
end;

```

Precizați care dintre următoarele șiruri de valori pot fi memorate în tabloul unidimensional **a** (cu indicii începând de la 0), astfel încât apelul **f(a,2,5)** să returneze valoarea 1.

1. a=(0,1,1,3,4,4,5)
2. a=(0,5,3,4,2,0)
3. a=(0,2,2,3,4,5,5)
4. a=(0,4,3,3,2,2,1)

- | | | |
|--------------------|--------------------|-----------------|
| a) niciunul | b) șirurile 1 și 3 | c) doar șirul 2 |
| d) șirurile 2 și 4 | e) doar șirul 3 | f) toate |

Varianta 10

1. Precizați ce valoare va avea variabila reală **x**, după executarea următoarei instrucțiuni.

Limbajul C++/ Limbajul C <code>x=7.51+35/4*67%8-2.83;</code>	Limbajul Pascal <code>x:=7.51+35 div 4*67 mod 8-2.83;</code>
---	---

a) instrucțiunea este incorectă b) 4 c) 4.68
d) 5 e) 28 f) 28.68

2. Precizați care dintre următoarele variante de instrucțiuni inserează cifra 2 în fața ultimei cifre a unui număr natural n.

Limbajul C++/ Limbajul C 1. <code>n=(n%10*10+2)*10+n/10;</code> 2. <code>n=(n/10*10+2)*10+n%10;</code> 3. <code>n=n/10+2*10+n%10;</code> 4. <code>n=(n/10+2)*10+n%10;</code>	Limbajul Pascal 1. <code>n:=(n mod 10*10+2)*10+n div 10;</code> 2. <code>n:=(n div 10*10+2)*10+n mod 10;</code> 3. <code>n:=n div 10+2*10+n mod 10;</code> 4. <code>n:=(n div 10+2)*10+n mod 10;</code>
--	---

a) niciuna b) 1 c) 1 și 2
d) 2 e) 3 f) 4

3. În secvența de program de mai jos, variabilele **x**, **y** și **z** sunt de tip întreg.

Limbajul C++/ Limbajul C <pre> z=1; while (y>0) { if (y%2) z=x%10*z; x=x*x%10; y=y/2; } cout<<z; printf("%d",z); </pre>	Limbajul Pascal <pre> z:=1; while y>0 do begin if y mod 2=1 then z:=x mod 10*z; x:=x*x mod 10; y:=y div 2; end; write(z); </pre>
---	--

Dacă înaintea secvenței, **x** are valoarea 137, precizați câte valori cu exact două cifre poate avea **y** astfel încât valoarea lui **z** (afișată la finalul secvenței) să fie 1.
a) 21 b) 22 c) 23
d) 24 e) 25 f) 26

4. Se consideră declarațiile:

Limbajul C++/ Limbajul C <pre> typedef struct { float st,dr; } interval; interval v[20], m; </pre>	Limbajul Pascal <pre> type interval=record st,dr:real; end; var v:array[0..19] of interval; m:interval; </pre>
---	---

`int i,j;`

`| i,j:word;`

Precizați care dintre următoarele instrucțiuni sunt corecte din punct de vedere sintactic.

Limbajul C++/ Limbajul C

```
1. v[i]=v[v[j]];
2. m=(v[2]+v[3])/2;
3. v[10]=m;
4. m.st=v[5].st%2;
```

Limbajul Pascal

```
1. v[i]:=v[v[j]];
2. m:=(v[2]+v[3])/2;
3. v[10]:=m;
4. m.st:=v[5].st mod 2;
```

a) niciuna

b) 1, 2 și 3

c) 1 și 4

d) 3

e) 3 și 4

f) toate

5. Subprogramul `f` cu antetul `int f(int x)` (în limbajul C++ și limbajul C), respectiv `function f(x:integer):integer;` (în limbajul Pascal), returnează cea mai mică cifră a numărului `x`, care apare de cel puțin două ori în scrierea lui `x`, sau valoarea -1, dacă numărul `x` este format din cifre distincte.

Stabiliți valoarea expresiei `f(f(775125) + f(97917))`.

a) -1

b) 0

c) 1

d) 12

e) 14

f) 16

6. Se generează prin metoda backtracking, submulțimile nevide ale mulțimii $\{1, 2, 3\}$ astfel: $\{1\}, \{1, 2\}, \{1, 2, 3\}, \{1, 3\}, \{2\}, \{2, 3\}, \{3\}$. Folosind aceeași metodă pentru a genera submulțimile nevide ale mulțimii $\{1, 2, 3, 4, 5, 6, 7\}$, stabiliți care este a 10-a, respectiv a 11-a soluție generată.

a) $\{1,2,3,4,7\},$
 $\{1,2,3,4,6,7\}$

b) $\{1,2,3,4,7\}, \{1,2,3,5\}$

c) $\{1,2,3,4,6,7\}, \{1,2,3,6,7\}$

d) $\{1,2,3,4,6\},$
 $\{1,2,3,4,6,7\}$

e) $\{1,2,3,4,6\}, \{1,2,3,5\}$

f) $\{1,2,3,4,6,7\}, \{1,2,3,4,7\}$

7. Se consideră graful neorientat $G=(X,U)$ unde $X=\{1,2,3,4,5\}$ și $U=\{[1,2], [1,3], [5,1], [3,4], [4,5], [3,2]\}$. Precizați numărul minim de muchii care trebuie adăugate pentru ca graful să devină eulerian.

a) 0

b) 1

c) 2

d) 3

e) 4

f) 5

8. Un graf neorientat are 30 noduri și 16 muchii. Precizați numărul componentelor conexe pe care le poate avea acest graf.

a) exact 14

b) cel puțin 14 și cel mult 24

c) cel puțin 14 și cel mult 26

d) cel puțin 16 și cel mult 24

e) cel puțin 16 și cel mult 26

f) exact 24

9. Fie $G=(V,E)$ un graf orientat în care mulțimea nodurilor este $V=\{1,2,\dots,10\}$, iar mulțimea arcelor este $E=\{(i,j) \mid V \times V \mid i \text{ este divizor propriu al lui } j\}$. Stabiliți care dintre următoarele afirmații este adevărată.

1. Graful G are 3 componente tare conexe

2. Graful G are 3 componente conexe

3. G nu are vârfuri izolate
4. Graful G conține cel puțin un circuit
- a) 1 b) 1 și 2 c) 2
d) 3 e) 4 f) toate
10. Prin înălțimea unui arbore cu rădăcină înțelegem numărul de muchii ale celui mai lung lanț elementar care are una dintre extremități în rădăcina arborelui. Dacă arborele este reprezentat prin următorul vector de tați: $tata = (4, 5, 1, 0, 4, 5, 6, 1, 4)$, atunci înălțimea sa este:
- a) 2 b) 3 c) 4
d) 5 e) 6 f) 7

11. Subprogramul **g** este definit mai jos. Precizați ce valori se vor afișa în urma apelul **g(4)**.

Limbajul C++/ Limbajul C

```
void g(int n)
{int i;
 if (n>0)
  for(i=n;i>1;i--)
   {
    cout<<i; | printf("%d",i);
    g(n-2);
   }
}
```

Limbajul Pascal

```
procedure g(n:word);
var i:word;
begin
 if n>0 then
  for i:=n downto 2 do
   begin
    write(i);
    g(n-2);
   end;
end;
```

- a) 42132 b) 42322 c) 423222
d) 432322 e) 432132121 f) 4213212211

12. În secvența de program alăturată, variabilele **i** și **k** sunt de tip întreg, iar variabila **A** memorează o matrice cu 8 linii și 8 coloane (numerotate de la 1 la 8) cu elemente numere întregi. Înainte de executarea secvenței, toate elementele vectorului sunt nule. Precizați care este a 9-a valoare afișată și care este numărul de valori afișate pe parcursul executării secvenței date.

Limbajul C++/ Limbajul C

```
n=8;k=3;
for (i=n;i>=1;i--)
  for (j=n;j>=1;j--)
    A[i][j]=n*(j-1)+i;
for(i=k;i<=n-k+1;i++)
  cout<<A[i][k]<<' '; | printf("%d ", A[i][k]);
for(i=k+1;i<=n-k+1;i++)
  cout<<A[n-k+1][i]<<' '; | printf("%d ",A[n-k+1][i]);
for(i=n-k;i>=k;i--)
  cout<<A[i][n-k+1]<<' '; | printf("%d ", A[i][n-k+1]);
for(i=n-k;i>k;i--)
  cout<<A[k][i]<<' '; | printf("%d ", A[k][i]);
```

Limbajul Pascal

```
n:=8;k:=3;
for i:=n downto 1 do
  for j:=n downto 1 do
```

```

        A[i,j]:=n*(j-1)+i;
    for i:=k to n-k+1 do
        write(A[i,k], ' ');
    for i:=k+1 to n-k+1 do
        write(A[n-k+1,i], ' ');
    for i:=n-k downto k do
        write(A[i,n-k+1], ' ');
    for i:=n-k downto k+1 do
        write(A[k,i], ' ');

```

- a) 43 și 10 b) 43 și 12 c) 44 și 10
 d) 44 și 12 e) 45 și 10 f) 45 și 12

13. Se consideră declarațiile:

Limbajul C++/ Limbajul C

```

char s[30]="bacaacbc";
char t[3][3]={"ab","ac","bc"};

```

Limbajul Pascal

```

var i:word;
s:string[30];
t:array[0..2]of string[3];

```

Precizați ce se va afișa după executarea secvenței de program de mai jos.

Limbajul C++/ Limbajul C

```

for(int i=0;i<3;i++)
    if (!strstr(s,t[i]))
        strcat(s,t[2-i]);
    else cout<<t[i];      | printf("%s",t[i]);
cout<<strlen(s);        | printf("%d",strlen(s));
cout<<s;                 | printf("%s",s);

```

Limbajul Pascal

```

s:='bacaacbc';t[0]:='ab';t[1]:='ac';t[2]:='bc';
for i:=0 to 2 do
    if pos(t[i],s)=0 then s:=s+t[2-i]
    else write(t[i]);
write(length(s));
write(s);

```

- a) secvența este incorectă b) acbc8bacaacbc c) acbc10bacaacbc
 sintactic
 d) bcac10bacaacbc e) acb10bacaacbc f) acbc10bacaacbc

14. Subprogramul **p** este definit mai jos. Variabila **a** memorează un vector cu 100 de elemente numere întregi, aflate pe poziții numerotate de la 0 la 99. Precizați câte elemente divizibile cu 3 conține tabloul unidimensional **a** după executarea apelului **p(6,a)**.

Limbajul C++/Limbajul C

```

void p(int k, int v[100])
{ int w[100];
  v[0]=w[0]=1;
  for(int i=1;i<=k;i++)
  {
    for(int j=0;j<=i;j++)
    {
      if(j==0) v[j]=1;
      else
        if(i==j) v[j]=1;
    }
  }
}

```

Limbajul Pascal

```

type vector=array[0..99]of integer;
var a:vector;
procedure p(k:integer; var v:vector);
var w:vector; i,j:integer;
begin
  v[0]:=1; w[0]:=1;
  for i:=1 to k do begin
    for j:=0 to i do
      if j=0 then v[j]:=1
      else

```

```

        else v[j]=w[j-1]+w[j];
    }
    for(int j=0;j<=i;j++)
        w[j]=v[j];
}

```

a)1
d)4

b)2
e) 5

```

        if i=j then v[j]:=1
        else v[j]:=w[j-1]+w[j];
    for j:=0 to i do w[j]:=v[j];
end;
end;

```

c)3
f) 6

15. Variabilele **x,n,i,b,c** memorează numere naturale. Dacă inițial variabila **b** memorează valoarea 3 și variabila **x** memorează inițial valoarea 0, indicați valoarea afișată de secvența de program următoare:

Limbajul C++/ Limbajul C

```

for(i=0;i<=99;i++)
{
    c=0,n=i;
    while(n>0)
    {
        c=c+n%b;
        n=n/b;
    }
    if(c>x) x=c;
}
cout<<x; | printf("%d",x);

```

a) 4
d) 7

b) 5
e) 8

Limbajul Pascal

```

for i:=0 to 99 do
begin
    c:=0;n:=i;
    while n>0 do
    begin
        c:=c+n mod b;
        n:=n div b;
    end;
    if c>x then x:=c;
end;
write(x);

```

c) 6
f) 9

Varianta 11

1. Precizați care este numărul elementelor egale cu **1** (pentru **limbajul C/C++**)/**true** (pentru **limbajul Pascal**) aflate pe diagonala principală în urma executării secvenței de mai jos, în care **a** este un tablou bidimensional cu **n** linii și **n** coloane, iar **i, j** sunt variabile de tip întreg.

Limbajul C++

```
for (i=n; i>=1; i--)
for (j=n; j>=1; j--)
    a[i][j]=(i==j);
```

Limbajul C

```
for (i=n; i>=1; i--)
for (j=n; j>=1; j--)
    a[i][j]=(i==j);
```

Limbajul Pascal

```
for i:=n downto 1 do
for j:=n downto 1 do
    a[i,j] := (i=j);
```

- a) nu poate fi calculat b) n^2 c) $n-1$ d) $n^2 - 1$ e) n f) $(n+1) * n/2$

2. Precizați care este complexitatea algoritmului de interclasare a două șiruri, cu **m** și respectiv **n** numere naturale ($n \leq m$).

a) $O(\max(m, n))$

b) $O(\min(m, n))$

c) $O(m+n)$

d) $O(m*n)$

e) $O(m \log_2(n))$

f) $O(n \log_2(m))$

3. Indicați care dintre următoarele secvențe de program calculează în variabila **nr**, în mod **corect** și **eficient** ca timp de executare, numărul cuburilor naturale perfecte mai mici sau egale decât **n** (număr natural cunoscut).

a)

Limbajul C++

```
int n, nr=0, i;
cin>>n;
for (i=1; i<=n; i++)
if (i*i*i<=n)
nr++;
cout<<nr;
```

Limbajul C

```
int n, nr=0, i;
scanf("%d", &n);
for (i=1; i<=n; i++)
if (i*i*i<=n)
nr++;
printf("%d", nr);
```

Limbajul Pascal

```
var n, i, nr: integer;
BEGIN
    readln(n);
    nr:=0;
    for i:=1 to n do
        if i*i*i<=n then
            nr:=nr+1;
    writeln(nr);
END.
```

b)

Limbajul C++

```
int n, nr=0, i, j;
cin>>n;
for (i=1; i<=n; i++)
for (j=1; j<=i; j++)
if (j*j*j==i)
nr++;
cout<<nr;
```

Limbajul C

```
int n, nr=0, j, i;
scanf("%d", &n);
for (i=1; i<=n; i++)
for (j=1; j<=i; j++)
if (j*j*j==i)
nr++;
printf("%d", nr);
```

Limbajul Pascal

```
var
n, nr, i, j: integer;
BEGIN
    readln(n); nr:=0;
    for i:=1 to n do
        for j:=1 to i do
            if j*j*j==i then
                nr:=nr+1;
    writeln(nr);
END.
```

c)		
Limbajul C++ <pre>int n,nr; cin>>n; nr=(int)exp(1/2.0* log(n)); cout<<nr;</pre>	Limbajul C <pre>int n,nr; scanf("%d",&n); nr=(int)exp(1/2.0* log(n)); printf("%d",nr);</pre>	Limbajul Pascal <pre>var n,nr:integer; BEGIN readln(n); nr:=trunc(exp(1/2* ln(n))); writeln(nr); END.</pre>
d)		
Limbajul C++ <pre>int n, nr; cin>>n; nr=n*n*(n+1)*(n+1)/4; cout<<nr;</pre>	Limbajul C <pre>int n, nr; scanf("%d",&n); nr=n*n*(n+1)*(n+1)/4; printf("%d",nr);</pre>	Limbajul Pascal <pre>var n, nr:integer; BEGIN readln(n); nr:=n*n*(n+1)*(n+1)div 4; writeln(nr); END.</pre>
e)		
Limbajul C++ <pre>int n, nr; cin>>n; nr=n*n*(n+1)*(n+1)/6; cout<<nr;</pre>	Limbajul C <pre>int n, nr; scanf("%d",&n); nr=n*n*(n+1)*(n+1)/6; printf("%d",nr);</pre>	Limbajul Pascal <pre>var n, nr:integer; BEGIN readln(n); nr:=n*n*(n+1)*(n+1)div 6; writeln(nr); END.</pre>
f)		
Limbajul C++ <pre>int n, nr; cin>>n; nr=(int)exp(1/3.0* log(n)); cout<<nr;</pre>	Limbajul C <pre>int n, nr; scanf("%d",&n); nr=(int)exp(1/3.0* log(n)); printf("%d",nr);</pre>	Limbajul Pascal <pre>var n, nr:integer; BEGIN readln(n); nr:=trunc(exp(1/3* ln(n))); writeln(nr); END.</pre>

4. Se definește funcția:

$$M_n^k = \begin{cases} M_{n-1}^k + k * M_{n-1}^{k-1} & \text{pentru } k > 0 \\ 1 & \text{pentru } k = 0 \end{cases}$$

Dacă se citesc numerele naturale **n, k (n ≥ k)** și se apelează funcția recursivă scrisă într-un limbaj de programare cunoscut (**C++/C sau Pascal**) care evaluează funcția definită mai sus, valoarea calculată reprezintă:

- | | | |
|---|---|--|
| a) produsul cartezian | b) numărul submulțimilor unei mulțimi cu n elemente | c) afișarea tuturor aranjamentelor mulțimii {1,2,...,n} luate câte k |
| d) numărul submulțimilor cu k elemente ale unei mulțimi cu n elemente | e) afișarea tuturor combinațiilor mulțimii {1,2,...,n} luate câte k | f) nici una dintre aceste variante |

5. Indicați care este efectul prelucrării secvenței de program de mai jos, pentru x, y numere naturale cu $x \leq y$:

Limbajul C++

```
int f(int x,int
y){
if(x==y) return x;
else if(x>y)
return x-1;
else
return
f(++x,--y);}
int main()
{int x, y;
cin>>x>>y;
cout<<f(x,y);
return 0;}
```

Limbajul C

```
int f(int x,int
y){
if(x==y) return x;
else if(x>y)
return x-1;
else
return
f(++x,--y);}
int main()
{int x, y;
scanf("%d%d",&x,&y);
printf("%d",f(x,y));
return 0;}
```

Limbajul Pascal

```
var x,y:integer;
function
f(x,y:integer):integ
er;
begin
if x=y then f:=x
else
if x>y then f:=x-1
else
f:=f(succ(x),pred(y))
end;
begin
readln(x,y);
writeln(f(x,y));
end.
```

S-a notat cu $[a]$ partea întreagă a numărului a și cu $|a-b|$, modulul diferenței $a-b$.

a) $x+y$ b) $[y-x]$ c) $|y-x|$ d) x^y e) $[(x+y)/2]$ f) y^x

6. Indicați care este valoarea inițială a variabilei a dacă, în urma executării secvenței următoare de program, s-a afișat valoarea 6.

Limbajul C++

```
int a=... ,n;
n=16327;
while(n!=0)
{
switch(n%10){
case 0: case 2: case
4:case 6: case 8:
a=a+n%2;break;
case 1: case 3: case
5:case 7: case 9:
a=a-n%2; break;}
n=n/10;
}
cout<<a<<endl;
```

Limbajul C

```
int a=... ,n;
n=16327;
while(n!=0)
{
switch(n%10){
case 0: case 2:
case 4:case 6:
case 8:
a=a+n%2;break;
case 1: case 3:
case 5:case 7:
case 9: a=a-n%2;
break;}
n=n/10;
}
printf("%d",a);
```

Limbajul Pascal

```
var n,a:integer;
BEGIN
a:=...;
n:=16327;
while n>0 do
begin
case n mod 10 of
0,2,4,6,8:
a:=a+n mod 2;
1,3,5,7,9:
a:=a-n mod 2;
end;
n:=n div 10;
end;
writeln(a);
END.
```

a) 8 b) nici una dintre variante c) 9 d) 13 e) 10 f) 5

7. Fie un graf neorientat conex cu **40** de noduri și **70** de muchii. Precizați numărul maxim de muchii care pot fi eliminate astfel încât graful să rămână conex:

a) 31 b) 30 c) 1 d) 0 e) 39 f) 35

8. Precizați numărul grafurilor orientate complete cu n noduri care pot fi construite:
a) $4^{n*(n-1)/2}$ b) $4^{n*(n+1)/2}$ c) $3^{n*(n+1)/2}$ d) $4^{n/2}$ e) $2^{n*(n-1)/2}$ f) $3^{n*(n-1)/2}$
9. Precizați care din următoarele șiruri de numere **nu** poate reprezenta gradele vârfurilor unui arbore.
a) 2 2 1 1 b) 4 1 1 2 1 1 c) 1 2 1
d) 1 1 1 2 1 5 e) 1 1 1 2 1 4 f) 1 2 1 2 1 3

10. Precizați care este valoarea produsă de subprogramului **f** definit mai jos, în urma apelului **f(s)**, atunci când parametrul **s** primește șirul de caractere: **123abc4567**

Limbajul C++

```
int f( char s[100])
{int i, nr=0, n=strlen(s);
for(i=0;s[i]>='0'&& s[i]<='9';i++)
    nr=nr*10+s[n-i-1]-'0';
return nr;}
```

Limbajul C

```
int f( char s[100])
{int nr=0,i,n;
n=strlen(s);
for(i=0;s[i]>='0'&& s[i]<='9';i++)
    nr=nr*10+s[n-i-1]-'0';
return nr;}
```

Limbajul Pascal

```
function f(s:String):longint;
var n,nr,i:longint ;
begin
n:=Length(s);nr:=0;i:=1;
while (s[i]>='0') and (s[i]<='9') do
begin
nr:=10*nr+ord(s[n-i+1])-ord('0');
i:=i+1;
end;f:=nr;
end;
```

- a) 1234567 b) 123 c) 456 d) 321 e) 765 f) 7654

11. Precizați care este suma calculată în urma apelului **f(1,n)** al subprogramului **f** definit mai jos, unde **n** este un număr natural cunoscut:

Limbajul C++

```
int f(int i, int n)
{
    if(i<=3*n)
        if(i%3!=0)
            return f(i+1,n);
        else
            return i+f(i+1,n);
    else return 0;
}
```

Limbajul C

```
int f(int i, int n)
{
    if(i<=3*n)
        if(i%3!=0)
            return f(i+1,n);
        else
            return i+f(i+1,n);
    else return 0;
}
```

Limbajul Pascal

```
function f(i,n:integer):integer;  
begin  
  if i<=3*n then  
    if i mod 3 =0 then  
      f:=f(i+1,n)+i  
    else f:=f(i+1,n)  
  else f:=0  
end;
```

a) $\sum_{i=1}^{3n} i$ b) $\sum_{i=1}^n 3 * i$ c) $\sum_{i=1}^n i$ d) $\sum_{i=1}^n (i + 3)$ e) $\sum_{i=1}^n i/3$ f) $\sum_{i=1}^{n/3} i$

12. Utilizând metoda backtracking, se generează toate posibilitățile de a forma șiruri din trei cuvinte distincte din mulțimea {examen, reușit, promovat, nota, felicitări}. Primele trei soluții sunt: **examen reușit promovat**; **examen reușit nota**; **examen reușit felicitări**. Indicați care este soluția generată înainte de **felicitări examen reușit**.

a) nota examen felicitări b) nota felicitări promovat c) felicitări examen promovat
d) examen promovat e) felicitări nota examen f) promovat examen felicitări

13. Dacă pentru variabila **a** se citește valoarea **11**, precizați câte valori pot fi citite pentru variabila **b** astfel încât în urma apelului **f(a,b)**, subprogramul **f** de finit mai jos să producă valoarea **10**.

Limbajul C++

```
int f(int a,int b)  
{  if(a==b)  
    return a%2;  
    else  
    return  
f(a,(a+b)/2)+f((a+b)/2+1,b); }
```

Limbajul C

```
int f(int a,int b)  
{  if(a==b)  
    return a%2;  
    else  
    return  
f(a,(a+b)/2)+f((a+b)/2+1,b); }
```

Limbajul Pascal

```
function f(a,b:integer):integer;  
begin  
  if a=b then f:=a mod 2  
  else  
    f:=f(a,(a+b) div 2) +f((a+b) div 2+1,b);  
end;
```

a) 2 b) 1 c) 0 d) 3 e) 4 f) 7

14. Indicați pentru ce valori ale lui **n** și **p**, valoarea variabilei întregi **x** este 164, în urma apelului **f(1)** al subprogramului **f** definit mai jos.

Limbajul C++

```
int s[100],n,p,x;  
void f(int k)  
{  int i;  
    if(k==p+1) x++;
```

Limbajul C

```
int s[100],n,p,x;  
void f(int k)  
{  int i;  
    if(k==p+1) x++;
```

```

else
if(k==1)
for(i=1;i<=n;i++) f(k+1);
else
for(i=1+s[k-1];i<=n;i++)
{ s[k]=i;f(k+1); }

```

```

else
if(k==1)
for(i=1;i<=n;i++) f(k+1);
else
for(i=1+s[k-1];i<=n;i++)
{ s[k]=i;f(k+1); }

```

Limbajul Pascal

```

type vector=array[1..100] of integer;
var n,p,x:integer;
s: vector;
procedure f(k:integer);
var i,i1:integer;
begin
if k=p+1 then x:=x+1
else
if k=1 then
for i1:=1 to n do
begin
s[k]:=i1; f(k+1);
end
else
for i:=1+s[k-1] to n do
begin
s[k]:=i; f(k+1);
end; end;

```

a) 82 și 2

b) 164 și 1

c) 328 și 4

d) 41 și 123

e) 41 și 1

f) 81 și 2

15. Precizați care este valoarea produsă de subprogramul **f**, definit mai jos, în urma apelului **f(1999,7)**.

Limbajul C++

```

int f(int x, int n)
{if(n)
if(n%2==0)
return (f(x,n/2)*f(x,n/2))%10;
else
return (x*f(x,n-1))%10;
else
return 1;}

```

Limbajul C

```

int f(int x, int n)
{if(n)
if(n%2==0)
return
(f(x,n/2)*f(x,n/2))%10;
else
return (x*f(x,n-1))%10;
else
return 1;}

```

Limbajul Pascal

```

function f(x,n:integer):integer;
begin
if n=0 then f:=1
else
if n mod 2=0 then

```

a) 798800599919 b) 9 c) nu poate fi calculată valoarea d) 1 e) 0 f) 28

Varianta 12

1. Precizați care este numărul muchiilor care trebuie adăugate, într-un graf neorientat conex care are n noduri și $n-1$ muchii (n număr natural cunoscut), pentru a deveni complet:

a) $n * (n-1) / 2$ b) $(n-1) * (n-2) / 2$ c) n
 d) $(n+1) * n / 2$ e) 0 f) $n-1$

2. În secvența următoare i , j și n sunt variabile întregi, iar a este un tablou bidimensional format din n linii și n coloane numerotate de la 1 la n . Indicați care este suma elementelor de pe diagonala secundară a tabloului a , în urma executării acestei secvențe.

Limbajul C++ <pre>for(i=1; i<=n; i++) for(j=1; j<=n; j++) a[i][j] = (i+j) % n;</pre>	Limbajul C <pre>for(i=1; i<=n; i++) for(j=1; j<=n; j++) a[i][j] = (i+j) % n;</pre>	Limbajul Pascal <pre>for i:=1 to n do for j:=1 to n do a[i,j]:=(i+j) mod n;</pre>
a) n b) $n-1$	c) nu se poate calcula	d) $(n+1) * n / 2$ e) n^2 f) $n * n$

3. Indicați care dintre următoarele secvențe de program calculează în variabila nr , în mod **corect** și **eficient** ca timp de executare, suma primelor n cuburi naturale perfecte nenule (n număr natural cunoscut).

a) Limbajul C++ <pre>int n, nr=0, i; cin>>n; for(i=1;i<=n;i++) if(i*i*i<=n) nr+=i*i*i; cout<<nr;</pre>	Limbajul C <pre>int n, nr=0, i; scanf("%d",&n); for(i=1;i<=n;i++) if(i*i*i<=n) nr+=i*i*i; printf("%d",nr);</pre>	Limbajul Pascal <pre>var n, nr:integer; BEGIN readln(n); nr:=0; for i:=1 to n do if i*i*i<=n then nr:=nr+i*i*i; writeln(nr); END.</pre>
---	--	--

b) Limbajul C++ <pre>int n, nr=0, i, j; cin>>n; for(i=1;i<=n;i++) for(j=1;j<=i;j++) if(j*j*j==i) nr+=i*i*i; cout<<nr;</pre>	Limbajul C <pre>int n, nr=0, i, j; scanf("%d",&n); for(i=1;i<=n;i++) for(j=1;j<=i;j++) if(j*j*j==i) nr+=i*i*i; printf("%d",nr);</pre>	Limbajul Pascal <pre>var n, i, j nr:integer; BEGIN readln(n);nr:=0; for i:=1 to n do for j:=1 to i do if j*j*j=i then nr:=nr+i*i*i; writeln(nr); END.</pre>
--	---	---

c) Limbajul C++ <pre>int n, nr;</pre>	Limbajul C <pre>int n, nr;</pre>	Limbajul Pascal <pre>var n, nr:integer;</pre>
--	--	---

<pre>cin>>n; nr=(int)exp(1/3.0* log(n)); cout<<nr;</pre>	<pre>scanf("%d",&n); nr=(int)exp(1/3.0* log(n)); printf("%d",nr);</pre>	<pre>BEGIN readln(n); nr:=trunc(exp(1/3* ln(n))); writeln(nr); END.</pre>
d) Limbajul C++ <pre>int n, nr; cin>>n; nr=n*n*(n+1)*(n+1)/4; cout<<nr;</pre>	Limbajul C <pre>int n, nr; scanf("%d",&n); nr=n*n*(n+1)*(n+1)/4; printf("%d",nr);</pre>	Limbajul Pascal <pre>var n, nr:integer; BEGIN readln(n); nr:=n*n*(n+1)*(n+1) div 4; writeln(nr); END.</pre>
e) Limbajul C++ <pre>int n, nr; cin>>n; nr=(int)exp(1/2.0* log(n)); cout<<nr;</pre>	Limbajul C <pre>int n, nr; scanf("%d",&n); nr=(int)exp(1/2.0* log(n)); printf("%d",nr);</pre>	Limbajul Pascal <pre>var n, nr:integer; BEGIN readln(n); nr:=trunc(exp(1/2* ln(n))); writeln(nr); END.</pre>
f) Limbajul C++ <pre>int n, nr; cin>>n; nr=n*n*(n+1)*(n+1)/6; cout<<nr;</pre>	Limbajul C <pre>int n, nr; scanf("%d",&n); nr=n*n*(n+1)*(n+1)/6; printf("%d",nr);</pre>	Limbajul Pascal <pre>var n, nr:integer; BEGIN readln(n); nr:=n*n*(n+1)*(n+1) div 6; writeln(nr); END.</pre>

4. Se definește recursiv o funcție care calculează numărul combinațiilor de **n** luate câte **k**

$$\text{astfel: } C_n^k = \begin{cases} \frac{n-k+1}{k} * C_n^{k-1} & \text{pentru } k > 0 \\ 1 & \text{pentru } k = 0 \end{cases}$$

Dacă se citesc numerele naturale **n, k (n ≥ k)** și se apelează funcția recursivă scrisă într-un limbaj de programare cunoscut (**C++/C/Pascal**) care evaluează funcția definită mai sus, precizați care este numărul apelurilor necesare pentru a calcula C_n^k :

- a) **n** b) **k** c) **k-1** d) **n+1** e) **n+k** f) **k+1**
5. Precizați numărul circuitelor care trec prin toate nodurile unui graf orientat tare conex cu **n** noduri (**n** număr natural cunoscut).
- a) **cel puțin 1** b) **exact 1** c) **0** d) **cel mult 1** e) **exact n-2** f) **exact n+1**

6. Precizați ce afișează următoarea secvență de program:

Limbajul C++

```
int a=0,n;
cin>>n;
do
{
switch(n%10)
{
case 0: case 2:
case 4: case 6:
case 8:
a=a-(n%2)*(n%10);
break;
case 1: case 3:
case 5: case 7: case
9:
a=a+(n%2)*(n%10);
break;
}
n=n/10;
} while(n!=0);
cout<<a<<endl;
```

Limbajul C

```
int a=0,n;
scanf("%d",&n);
do
{
switch(n%10)
{
case 0: case 2:
case 4: case 6:
case 8:
a=a-(n%2)*(n%10);
break;
case 1: case 3:
case 5: case 7: case
9:
a=a+(n%2)*(n%10);
break;
}
n=n/10;
} while(n!=0);
printf("%d",a);
```

Limbajul Pascal

```
var n,a:integer;
BEGIN
a:=0;
readln(n);
while n>0 do
begin
case n mod 10 of
0,2,4,6,8:
a:=a-(n mod 2)*
(n mod 10);
1,3,5,7,9:
a:=a+(n mod 2)*
(n mod 10);
end;
n:=n div 10;
end;
writeln(a);
END.
```

- a) suma cifrelor numărului **n** b) numărul cifrelor impare din numărul **n** c) diferența dintre suma cifrelor pare și suma cifrelor impare din numărul **n**
d) numărul cifrelor prime din numărul **n** e) suma cifrelor impare din numărul **n** f) diferența dintre suma cifrelor impare și suma cifrelor pare din numărul **n**

7. Folosind metoda backtracking se generează toate șirurile formate din patru caractere distincte din mulțimea {#,*,&,@,%}. Primele trei soluții sunt: #*&@, #*&%, #*%@&. Indicați care este soluția generată înainte de &*#%

- a) &@%# b) &#%@ c) &#%@ d) *%&@ e) *%&@ f) &@%#

8. Un graf orientat se numește turneu, dacă între oricare două vârfuri **i** și **j**, **i ≠ j**, există un singur arc. Precizați numărul grafurilor turneu cu **n** noduri (**n** număr natural cunoscut).

- a) $4^{n*(n-1)/2}$ b) $3^{n*(n-1)/2}$ c) $4^{n*(n+1)/2}$
d) $2^{n*(n-1)/2}$ e) $2^{n*(n+1)/2}$ f) $3^{n*(n+1)/2}$

9. Precizați care din următoarele șiruri de numere poate reprezenta gradele vârfurilor unui arbore cu **n** noduri (**n** număr natural cunoscut).

- a) 4 1 1 2 1 1 b) 2 1 1 1 c) 1 2 2
d) 1 1 1 2 1 5 e) 1 1 1 f) 1 1 1 2 1 6

10. Precizați care este valoarea produsă de subprogramul **f**, definit mai jos, în urma apelului **f(s)**, atunci când variabila **s** memorează șirul de caractere: **123abc45678**

Limbajul C++

```
int f( char s[100])
{int nr=0,i,n=strlen(s),p=1;
for(i=0;s[i]>='0'&& s[i]<='9'; i++)
    {nr=nr+p*(s[n-i-1]-'0');
    p*=10;}
return nr;
}
```

Limbajul C

```
int f( char s[100])
{int nr=0,i,
n=strlen(s),p=1;
for(i=0;s[i]>='0'&& s[i]<='9'; i++)
    {nr=nr+p*(s[n-i-1]-'0');
    p*=10;}
return nr;}
```

Limbajul Pascal

```
function f(s:String):longint;
var n,nr,i,p:longint;
begin
n:=Length(s);nr:=0;i:=1;p:=1;
while (s[i]>='0') and (s[i]<='9') do
begin
nr:=nr+p*(ord(s[n-i+1])-ord('0'));
i:=i+1; p:=p*10;
end;f:=nr;
end;
```

- a) 45678 b) 123 c) 876 d) 678 e) 654 f) 876

11. În urma apelului **f(1,n)** al subprogramului **f** definit mai jos, este calculată valoarea sumei...:

Limbajul C++

```
float f(int i,int
n)
{ if(i<=3*n)
    if(i%3!=0)
        return f(i+1,n);
    else
return
1.0/i+f(i+1,n);
    else return 0;}
```

Limbajul C

```
float f(int i,int
n)
{ if(i<=3*n)
    if(i%3!=0)
        return f(i+1,n);
    else
return
1.0/i+f(i+1,n);
    else return 0;}
```

Limbajul Pascal

```
function
f(i,n:integer):double;
begin
if i<=3*n then
if i mod 3 = 0 then
f:=f(i+1,n)+1/i
else f:=f(i+1,n)
else f:=0
end;
```

- a) $\sum_{i=1}^{3n} 1/i$ b) $\sum_{i=1}^n 1/(3*i)$ c) $\sum_{i=1}^n 1/i$ d) $\sum_{i=1}^{2n} 1/i$ e) $\sum_{i=1}^n 1/(i-3)$ f) $\sum_{i=1}^n 1/(i+3)$

12. Precizați care este valoarea variabilei **x**, în urma apelului **f(1)** al subprogramului **f** definit mai jos, pentru **n=4** și **p=3**.

Limbajul C++

```
int s[100],n,p,x;
void f(int k)
{ int i,j,ok;
if(k==p+1)x++;
    else
for(i=1;i<=n;i++)
{s[k]=i;
ok=1;
```

Limbajul C

```
int s[100],n,p,x;
void f(int k)
{int i,j,ok;
if(k==p+1) x++;
    else
for(i=1;i<=n;i++)
{s[k]=i;
ok=1;
```

Limbajul Pascal

```
type
vector=array[1..100]
of integer;
var n,p,x:integer;
s: vector;

procedure
f(k:integer);
```

```
for (j=1; j<k; j++)
if (s[k]==s[j])
ok=0;
if (ok) f(k+1);
}}
```

```
for (j=1; j<k; j++)
if (s[k]==s[j])
ok=0;
if (ok) f(k+1);
}}
```

```
var i, i1, j, ok: integer;
begin
if k=p+1 then
x:=x+1
else
if k=1 then
for i1:=1 to n do
begin
s[k]:=i1; f(k+1);
end
else
for i:=1 to n do
begin
s[k]:=i;
ok:=1;
for j:=1 to k-1 do
if s[k]=s[j] then
ok:=0;
if ok=1 then f(k+1);
end; end;
```

a) 120

b) 12

c) 10

d) 8

e) 24

f) 60

13. Precizați numărul maxim de muchii care pot exista într-un graf neorientat cu n noduri (n număr natural cunoscut):

a) n b) $n*(n+1)/2$ c) $n*(n-1)/2$ d) $n*n*(n-1)$ e) $n+1$ f) $n-1$

14. Dacă pentru variabila n se citește valoarea 12, precizați care este valoarea produsă în urma apelului $f(1, n, n)$ al subprogramului f definit mai jos:

Limbajul C++

```
int f(int a, int b, int n)
{
if (a==b)
return (n%a==0?1:0);
else
return
f(a, (a+b)/2, n)+f((a+b)/2+1, b, n); }
```

Limbajul C

```
int f(int a, int b, int n)
{
if (a==b)
return (n%a==0?1:0);
else
return
f(a, (a+b)/2, n)+f((a+b)/2+1, b, n); }
```

Limbajul Pascal

```
function f(a, b, n: integer): integer;
begin
if a=b then
if n mod a=0 then f:=1
else f:=0
else f:=f(a, (a+b) div 2, n)+f((a+b) div 2+1, b, n);
end;
```

a) 0

b) 2

c) 4

d) 6

e) 12

f) 8

15. Indicați ce reprezintă valoarea afișată în urma rulării programului de mai jos:

Limbajul C++

```
#include<iostream>
using namespace std;
int f(int n, int k)
{if(k==1||k==n) return 1;
  else
    if(k>n)return 0;
  else
    return
      f(n-1,k-1)+k*f(n-1,k); }
int main()
{  int n, k,s=0;
  cin>>n;
  for(k=1;k<=n;k++)
    s=s+f(n,k);
  cout<<s; return 0;}
```

Limbajul C

```
#include <stdio.h>
int f(int n, int k)
{if(k==1||k==n) return 1;
  else
    if(k>n)return 0;
  else
    return
      f(n-1,k-1)+k*f(n-1,k); }
int main()
{  int n, k,s=0;
  scanf("%d",&n);
  for(k=1;k<=n;k++)
    s=s+f(n,k);
  printf("%d",s); return 0;}
```

Limbajul Pascal

```
function f(n,k:integer):integer;
begin
  if k=1 then f:=1
  else if k=n then f:=1
  else if k>n then f:=0
  else
    f:=f(n-1,k-1)+k*f(n-1,k);
  end;
var n,k,s:integer;
begin
  readln(n);s:=0;
  for k:=1 to n do s:=s+f(n,k);
  writeln(s);readln;
end.
```

a) A_n^k

d) numărul submulțimilor
mulțimii $\{1,2,...n\}$

b) numărul submulțimilor
ordonate mulțimii $\{1,2,...n\}$

e) C_n^k

c) numărul total al
partițiilor mulțimii $\{1,2,...n\}$

f) numărul submulțimilor
ordonate mulțimii $\{1,2,...k\}$

Varianta 13

1. Indicați câte dintre expresiile următoarele au valoarea **1** (**Limbajul C/C++**), respectiv **true** (**Limbajul Pascal**) dacă și numai dacă valorile variabilelor **a** și **b** sunt numere întregi pare consecutive.

Limbajul C/C++

1. `(a%2) && (b%2) && (a-b==2)`
2. `(a%2) && (a-b==2 || b-a==2)`
3. `!(a%2) && abs(a-b)==2`
4. `!(a%2) && !(b%2) && abs(a-b)==2`
5. `!!(a%2) && (a-b==2)`
6. `(a%2==0) && !(abs(a-b)==2)`

a) 0

b) 1

c) 2

Limbajul Pascal

1. `(a mod 2<>0) and (b mod 2<>0) and (a-b=2)`
2. `(a mod 2<>0) and ((a-b=2) or (b-a=2))`
3. `not(a mod 2<>0) and (abs(a-b)=2)`
4. `not(a mod 2<>0) and not(b mod 2<>0) and (abs(a-b)=2)`
5. `not(not(a mod 2<>0)) and (a-b=2)`
6. `(a mod 2=0) and not(abs(a-b)=2)`

d) 3

e) 4

f) 5

2. Precizați ce se va afișa în urma rulării secvenței următoare, în care se consideră că variabilele **a** și **b** memorează numere reale.

Limbajul C++

```
a=5.2;
b=-3.25;
a-=b;
b*=2;
cout<<ceil(a+b)<<" "<<
floor(a-b);
```

Limbajul C

```
a=5.2;
b=-3.25;
a-=b;
b*=2;
printf("%g %g",ceil(a+b),floor(a-b));
```

a) -5 8

b) -4 8

c) 1 14

d) 1 15

e) 2 14

f) 2 15

Limbajul Pascal

```
a:=5.2;
b:=-3.25;
a:=a-b;
b:=b*2;
write(round(a+b),' ',trunc(a-b));
```

3. Precizați ce se afișează la sfârșitul executării secvenței următoare.

Limbajul C++

```
void p(int a, int &b)
{ a++;
  b=b*a;
  b-=10;}
int g(int a, int b)
```

Limbajul Pascal

```
var a,b:integer;
procedure p(a:integer; var
b:integer);
begin
  inc(a);
```

```

{ a*=10;
  b+=a;
  a=b;
  return a;}
int main()
{ int a=2,b=7;
  p(a,b);
  cout<<g(b,a);}

```

Limbajul C

```

void p(int a, int *b)
{ a++;
  *b=*b*a;
  *b-=10;}
int g(int a, int b)
{ a*=10;
  b+=a;
  a=b;
  return a;}
int main()
{ int a=2,b=7;
  p(a,&b);
  printf("%d",g(b,a)); }

```

a) 27

b) 41

c) 72

```

      b:=b*a;
      b:=b-10;
end;

function
g(a,b:integer):integer;
begin
  a:=a*10;
  b:=b+a;
  a:=b;
  g:=a;
end;

begin
  a:=2;
  b:=7;
  p(a,b);
  write(g(b,a));
end.

```

d) 73

e) 112

f) 113

4. Se consideră subprogramul **f** definit mai jos. Precizați ce se afișează în urma apelului **f(8)**.

Limbajul C++

```

void f(int i)
{ if (i>1)
  if (i%2)
    {f(i-1);
     cout<<i-1<<" ";}
  else {i--;
        f(i);}
}

```

Limbajul C

```

void f(int i)
{ if (i>1)
  if (i%2)
    {f(i-1);
     printf("%d ",i-1);}
  else {i--;
        f(i);}
}

```

a) 2 3 5

b) 2 4 6

c) 3 5 7

Limbajul Pascal

```

procedure f(i:integer);
begin
  if i>1 then
    if i mod 2<>0 then
      begin
        f(i-1);
        write(i-1,' ');
      end
    else
      begin
        dec(i);
        f(i);
      end;
end;

```

d) 5 3 2

e) 6 4 2

f) 7 5 3

5. Precizați ce se va afișa în urma rulării secvenței date, în care se consideră că variabilele **x** și **y** sunt de tip întreg.

Limbajul C++

```
x=5;
y=2;
cout<<+x/y+++1;
cout<<endl<<x<<" "<<y;
```

Limbajul Pascal

```
x:=5;
y:=2;
inc(x);
writeln(x div y+1);
inc(y);
writeln(x,' ',y);
```

Limbajul C

```
x=5;
y=2;
printf("%d",++x/y+++1);
printf("\n%d %d",x,y);
```

- a) 4 b) 4 c) 4 d) 3 e) 3 f) 3
 6 6 2 5 2 6 3 6 2 5 2
 3

6. Știind că variabila **s** este de tip șir de caractere, precizați ce se va afișa după executarea următoarei secvențe de instrucțiuni.

Limbajul C/C++

```
strcpy(s, "ExamenUPB");
for (i=0;i<strlen(s)/2;i++)
    s[i]=s[strlen(s)-i-2];
strcpy(s,s+2);
strcpy(s+strlen(s)-2,
s+strlen(s)-1);
printf("%s",s); | cout<<s;
```

Limbajul Pascal

```
s:='ExamenUPB';
for i:=1 to length(s) div 2
do
s[i]:=s[length(s)-i];
delete(s,1,2);
delete(s,length(s)-1,1);
write(s);
```

- a) UnenUB b) UnenUP c) neenUB d) neenUP e) nennUB f) nennUP

7. Indicați care este numărul de comparații executate pentru ordonarea descrescătoare a unui tablou unidimensional cu **50** elemente, prin metoda interschimbării.

- a) 25 b) 49 c) 50 d) 1225 e) 1226 f) 2450

8. Utilizând metoda backtracking se generează toate codurile formate din cinci caractere distincte ale mulțimii **{a,b,c,d,e,f}**. Primele cinci soluții generate sunt: **abcde, abcdf, abced, abcef, abcfed**. Indicați care sunt codurile generate imediat în fața soluției **dcbae**, dar și imediat după aceasta.

- a) dcaef; b) dcafe; c) dcbafe; d) dcbfe; e) dcbef; f) dcbfa;
 dcafe dcbafe dcbea dceab dcbfe dcbaf

9. La o cantină se prepară zilnic **5** sortimente pentru felul întâi, **10** pentru felul doi și **6** tipuri de desert. Precizați câte posibilități de a alege un meniu există, știind că un meniu este alcătuit din felul întâi, felul doi și facultativ desert.

- a) 50 b) 80 c) 90 d) 250 e) 300 f) 350
10. Fie un graf neorientat cu **10** noduri. Gradele vârfurilor acestuia sunt reținute în șirul: **4, 2, 2, 3, 3, 3, 2, 4, 2, 3**. Precizați care este numărul de muchii ce trebuie adăugate pentru ca graful să devină complet.
a) 45 b) 41 c) 31 d) 27 e) 26 f) 17
11. Fie un graf neorientat complet cu **50** noduri. Precizați care este numărul minim de muchii care trebuie eliminate pentru ca graful să fie hamiltonian.
a) 0 b) 25 c) 50 d) 612 e) 1175 f) 1225
12. Precizați câți arbori binari cu **3** noduri, numerotate de la **1** la **3**, se pot construi. Un arbore binar este un arbore în care fiecare nod are cel mult doi descendenți direcți (fii), ordonați: fiu stâng, fiu drept. Dacă un nod are un singur descendent trebuie specificat dacă este fiu stâng sau fiu drept.
a) 6 b) 8 c) 9 d) 21 e) 24 f) 30
13. Fie arborele cu rădăcină cu nodurile numerotate de la **1** la **15**, reprezentat prin vectorul de tați: **(10, 8, 4, 10, 1, 4, 5, 10, 8, 0, 3, 5, 3, 12, 3)**. Precizați câți descendenți are nodul **4**.
a) 6 b) 5 c) 4 d) 3 e) 2 f) 1
14. Precizați câte grafuri neorientate distincte cu **25** noduri, dintre care cel puțin un nod este izolat, se pot construi.
a) 2^{276} b) 2^{300} c) $3 \cdot 2^{279}$ d) $5^2 \cdot 2^{276}$ e) $3 \cdot 2^{303}$ f) $5^2 \cdot 2^{300}$
15. Fie secvența de instrucțiuni folosită pentru ridicarea la puterea **p** a unei matrice pătratică **a**, de ordin **n**. Elementele tabloului **a** sunt numere întregi, iar **n** și **p** sunt numere naturale nenule. Variabilele **a**, **b**, **c** sunt tablouri bidimensionale, cu **n** linii și **n** coloane, iar variabilele **i**, **j**, **k** sunt de tip întreg.

Limbajul C/C++

```
for (i=1; i<=n; i++)
    for (j=1; j<=n; j++)
        b[i][j]=(i==j);
for (q=1; q<=p; q++)
{for (i=1; i<=n; i++)
    for (j=1; j<=n; j++)
        {c[i][j]=0;
         for (k=1; k<=n; k++)
             c[i][j]=
c[i][j]+b[i][k]*a[k][j];
        }
for (i=1; i<=n; i++)
    for (j=1; j<=n; j++)
        b[i][j]=c[i][j];
```

Limbajul Pascal

```
for i:=1 to n do
    for j:=1 to n do
        if i=j then b[i,j]:=1
        else b[i,j]:=0;
for q:=1 to p do
    begin
        for i:=1 to n do
            for j:=1 to n do
                begin
                    c[i,j]:=0;
                    for k:=1 to n do
                        c[i,j]:=c[i,j]+b[i,k]*a[k,j];
                    end;
                for i:=1 to n do
```

```

}                                     for j:=1 to n do
                                     b[i,j]:=c[i,j];
                                     end;

```

Precizați de câte ori se execută operația de înmulțire în cadrul secvenței date pentru ridicarea la puterea p a matricei pătratice a de ordin n .

- a) $(n^3)^p$ b) $(p+1) \cdot n^3$ c) $p \cdot n^3$ d) $(p-1) \cdot n^3$ e) $p \cdot n^2$ f) n^3

Varianta 14

1. Indicați care expresie dintre următoarele are valoarea **1** (**Limbajul C/C++**), respectiv **true** (**Limbajul Pascal**) dacă și numai dacă valorile variabilelor **a** și **b** sunt numere întregi impare consecutive.

Limbajul C/C++

1. `(a%2==1) &&! (b%2) && abs(a-b)==2`
2. `! (a%2) &&! (b%2) && abs(a-b)==2`
3. `! (! (a%2) &&! (b%2) ) && abs(a-b)==2`
4. `(a%2==1&&b%2!=1) && (a-b==2 || b-a==2)`
5. `! (a%2) &&! (b%2) &&! (abs(a-b)==2)`
6. `(a%2) &&(b%2) && abs(a-b)!=2`

a) 1

b) 2

c) 3

Limbajul Pascal

1. `(a mod 2=1) and (not(b mod 2<>0)) and (abs(a-b)=2)`
2. `not(a mod 2<>0) and not(b mod 2<>0) and (abs(a-b)=2)`
3. `not(not(a mod 2<>0) and (not(b mod 2<>0))) and (abs(a-b)=2)`
4. `((a mod 2=1) and (b mod 2<>1)) and ((a-b=2) or (b-a=2))`
5. `not(a mod 2<>0) and not(b mod 2<>0) and not(abs(a-b)=2)`
6. `(a mod 2<>0) and (b mod 2<>0) and (abs(a-b)<>2)`

d) 4

e) 5

f) 6

2. Precizați ce se va afișa în urma rulării secvenței următoare, dacă variabila întreagă **x** are valoarea inițială **1234**.

Limbajul C++

```
x=x%100/10*10/10%10 +
x/10%10;
cout<<ceil(sqrt(x)+0.5)<<
" "<< floor(sqrt(x)-0.5);
```

Limbajul C

```
x=x%100/10*10/10%10 +
x/10%10;
printf("%g %g",
ceil(sqrt(x)+0.5),
floor(sqrt(x)-0.5));
```

a) 2 0

b) 2 1

c) 2 2

Limbajul Pascal

```
x:= x mod 100 div 10*10 div
10 mod 10 +x div 10 mod 10;
write(round(sqrt(x)+0.5),
' ',trunc(sqrt(x)-0.5));
```

d) 3 1

e) 3 2

f) 5 3

3. Precizați ce se afișează la sfârșitul executării secvenței următoare.

Limbajul C++

```
void p(int a, int &b)
{ a+=a+b;
  b+=a;
  a=b-a; }
```

Limbajul Pascal

```
var a,b:integer;
procedure p(a:integer; var
b:integer);
begin
```

```
int main()
{ int a=5,b=10;
  p(a,b);
  cout<<a<<" "<<b;
  p(a,b);
  cout<<endl<<a<<" "<<b;
}
```

Limbajul C

```
void p(int a, int *b)
{ a+=a>(*b);
  *b+=a;
  a=*b-a; }
int main()
{ int a=5,b=10;
  p(a,&b);
  printf("%d %d",a,b);
  p(a,&b);
  printf("\n%d %d",a,b);}
```

a) 5 30 b) 5 30 c) 5 30
5 70 5 80 10 80

```
a:=a+a+b;
b:=b+a;
a:=b-a;
end;

begin
a:=5;
b:=10;
p(a,b);
writeln(a,' ',b);
p(a,b);
writeln(a,' ',b);
end.
```

d) 10 30 e) 10 30 f) 10 30
10 30 30 70 30 80

4. Precizați care este valoarea returnată de funcție la apelul **f(1502)**.

Limbajul C/C++

```
int f(int i)
{ if(i==0) return 10;
  else
    if(i%10==0 || i%10==5)
      return f(i/10)*10+i;
    else
      return i%10 * f(i/10);
}
```

a) 3175 b) 2600 c) 2330

Limbajul Pascal

```
function
f(i:integer):integer;
begin
  if i=0 then f:=10
  else
    if (i mod 10=0) or (i mod 5=0)
      then f:=f(i div 10)*10 + i
    else
      f:=(i mod 10)*f(i div 10);
end;
```

d) 2050 e) 530 f) 350

5. Precizați ce se va afișa în urma rulării secvenței următoare, în care se consideră că variabilele **x** și **y** sunt de tip întreg.

Limbajul C++

```
x=2; y=5;
while(x<y)
{ cout<<++x+y++<<" ";
  x++; }
```

Limbajul C

```
x=2; y=5;
while(x<y)
```

Limbajul Pascal

```
x:=2; y:=5;
while x<y do
begin
  inc(x);
  write(x+y,' ');
  inc(y);
  inc(x);
end;
```

```
{ printf("%d ", ++x+y++);
  x++; }
```

a) 7 10 13

b) 8 10

c) 8 11 14

d) 9 11

e) 9 12

f) 9 12 15

6. Știind că variabila **s** este de tip șir de caractere, precizați ce se va afișa după executarea următoarei secvențe de instrucțiuni.

Limbajul C/C++

```
strcpy(s, "Examen-UPB");
for(i=strlen(s)/2; i>0; i--)
    s[i]=s[strlen(s)-i];
strcpy(s+strlen(s)/2-1,
s+strlen(s)/2+1);
printf("%s", s); | cout<<s;
```

Limbajul Pascal

```
s:='Examen-UPB';
for i:=length(s) div 2 downto 2
do s[i]:=s[length(s)-i+2];
delete(s, length(s) div 2, 2);
write(s);
```

a) -UPB

b) -n-UPB

c) n-UPB

d) EBPUPB

e) EBPUPB

f) EBPUPB

7. Indicați care este numărul necesar de comparații pentru ordonarea prin interschimbare a unui tablou unidimensional cu **100** elemente.

a) 99

b) 2475

c) 4851

d) 4950

e) 5050

f) 10000

8. Indicați câte numere divizibile cu **10**, cu **10** cifre, pot fi construite folosind numai cifrele **0, 1 și 2**.

a) 6561

b) 13122

c) 13212

d) 15322

e) 19683

f) 59049

9. Fie mulțimile **A={1, 2, 3, 4}**, **B={1, 2, 3}**, **C={1, 2}**, **D={1, 2, 3, 4}**. Precizați care este al **10**-lea element al produsului cartezian **A×B×C×D**, cât și antepenultimul element.

a) 1212; 4322

b) 1212; 4323

c) 1213; 4322

d) 1221; 4322

e) 1312; 4322

f) 1312; 4323

10. Fie un graf neorientat cu **25** noduri și **40** muchii. Precizați care este numărul maxim de noduri izolate pe care le poate avea graful.

a) 16

b) 15

c) 14

d) 13

e) 10

f) 5

11. Fie un graf neorientat cu **100** noduri. Precizați care este numărul minim de muchii necesar pentru ca graful să nu aibă noduri izolate.

a) 48

b) 49

c) 50

d) 98

e) 99

f) 100

12. Precizați care este numărul maxim de frunze al unui arbore binar cu **100** noduri care are înălțimea minimă. Un arbore binar este un arbore în care fiecare nod are cel mult doi descendenți direcți (fii).

a) 99

b) 69

c) 51

d) 50

e) 37

f) 34

13. Fie arborele cu rădăcină cu nodurile numerotate de la 1 la 15, reprezentat prin vectorul de tați: (10, 8, 4, 10, 1, 4, 5, 10, 8, 0, 3, 5, 3, 12, 3). Precizați câte lanțuri elementare distincte de lungime 3, care pleacă din rădăcină, există.

a) 2 b) 3 c) 4 d) 5 e) 6 f) 7

14. Precizați câte grafuri orientate distincte cu 25 noduri, dintre care cel puțin un nod este izolat, se pot construi.

a) $5^2 \cdot 2^{600}$ b) $3 \cdot 2^{603}$ c) 2^{600} d) $5^2 \cdot 2^{552}$ e) $3 \cdot 2^{555}$ f) 2^{552}

15. Precizați care este complexitatea timp pentru următoarea secvență de program, unde n reprezintă numărul de elemente al unui tablou unidimensional v , numerotat de la 1 la n , cu elemente numere întregi, iar x un număr întreg.

Limbajul C/C++

```
j=0;
for(i=1; i<=n; i++)
    if(v[i]!=x)
        { j++; v[j]=v[i]; }
n=j;
```

Limbajul Pascal

```
j:=0;
for i:=1 to n do
    if v[i]<>x then
        begin
            inc(j); v[j]:=v[i];
        end;
n:=j;
```

a) $O(n)$
d) $O(n^2)$

b) $O(\log n)$
e) $O(n^3)$

c) $O(n \cdot \log n)$
f) $O(2^n)$

Varianta 15

1. Se consideră variabilele de tip întreg $a=300$, $b=5$, $c=3$, $d=2$ și R , indicați valoarea variabilei R în urma executării instrucțiunii:

Limbajul C/C++ $R=a/b*c/d$; **Limbajul Pascal** $R:=a \text{ div } b * c \text{ div } d$;

a) 50 b) 10 c) 40 d) 60 e) 5 f) 90

2. Fie următoarele două secvențe de cod:

Limbajul C/C++

Secvența 1:

```
s=0;
for (i=1; i<=n; i++)
s=s+i*i;
```

Secvența 2:

```
s=0; i=<initial>;
while (<condition>)
{ s=s+i*i; i=i-1;
}
```

Limbajul Pascal

Secvența 1:

```
s:=0;
for i:=1 to n do s:=s+i*i;
```

Secvența 2:

```
s:=0; i:=<initial>;
while <condition> do
begin
s:=s+i*i; i:=i-1;
end;
```

Cu ce trebuie înlocuite **<initial>** și **<condition>** astfel încât cele două secvențe de cod să fie echivalente (în final variabila **s** să aibă aceeași valoare)?

a) 1 și $i \leq n$ b) n și $i > 0$ c) 1 și $i < n$ d) n și $i > 1$ e) 1 și $i \leq n-1$ f) $n-1$ și $i > 1$

3. Precizați secvența de instrucțiuni echivalentă cu următoarea secvență de cod.

Limbajul C/C++

```
if (a>b)
    if (a%2==0) c=a;
    else c=b;
else
    if (b%2==0) c=a;
    else c=b;
```

Limbajul Pascal

```
if a>b then
begin
    if a mod 2=0 then c:=a
    else c:=b;
end
else
    if b mod 2=0 then c:=a
    else c:=b;
```

Limbajul C/C++

- a) `if (a>b && a%2==0 || b>=a && b%2==0) c=a; else c=b;`
 b) `if (a>b && a%2==0 || b%2==0) c=a; else c=b;`
 c) `if (a>b && a%2==0 && b>=a && b%2==0)`

Limbajul Pascal

- a) `if ((a>b) and (a mod 2=0)) or ((b>=a) and (b mod 2=0)) then c:=a else c:=b;`
 b) `if ((a>b) and (a mod 2=0)) or (b mod 2=0) then c:=a else c:=b;`
 c) `if ((a>b) and (a mod 2=0)) or (b mod 2=0) then c:=a else c:=b;`

<pre> c=a; else c=b; d) if(a>b a%2==0 && b%2==0) c=a; else c=b; e) if(a>b a%2==0) c=a; else c=b; f) if(a%2==0 && b%2==0) c=a; else c=b; </pre>	<pre> if (a>b) and (a mod 2=0) and (b>=a) and (b mod 2=0) then c:=a else c:=b; d) if ((a>b) or (a mod 2=0)) and (b mod 2=0) then c:=a else c:=b; e) if (a>b) or (a mod 2=0) then c:=a else c:=b; f) if (a mod 2=0) and (b mod 2=0) then c:=a else c:=b; </pre>
---	---

4. Indicați numărul de muchii ce trebuie adăugate într-un graf neorientat care are **8** noduri și **20** muchii, astfel încât acesta să devină complet.

a) 20 b) 0 c) 4 d) 28 e) 8 f) 30

5. Precizați care dintre următoarele șiruri de numere pot reprezenta vectorul de tați al unui arbore binar cu rădăcină. Se numește arbore binar un arbore în care fiecare nod are maxim doi descendenți direcți.

a) (5,8,4,0,4,5,3,2,7,8) b) (5,1,4,0,4,5,3,1,7,1) c) (5,8,4,0,4,5,3,6,7,8)
d) (5,8,4,0,4,9,3,6,2,7) e) (5,0,4,0,4,5,3,1,7,1) f) (2,1,4,0,4,5,3,1,7,1)

6. Se consideră un tablou bidimensional **A**, cu **n** linii și **n** coloane, notăm cu **A_{ij}** elementul aflat pe linia **i** și coloana **j** ($1 \leq i \leq n, 1 \leq j \leq n$). Precizați condiția necesară ca elementul **A_{ij}** să fie situat deasupra ambelor diagonale.

Limbajul C/C++

a) **i<j && i+j<n+1**
b) **i>j && i+j>n+1**
c) **i<j || i+j<n+1**
d) **i<j && i+j>n+1**
e) **i>j || i+j>n+1**
f) **i<j || i+j>n+1**

Limbajul Pascal

a) **(i<j) and (i+j<n+1)**
b) **(i>j) and (i+j>n+1)**
c) **(i<j) or (i+j<n+1)**
d) **(i<j) and (i+j>n+1)**
e) **(i>j) or (i+j>n+1)**
f) **(i<j) or (i+j>n+1)**

7. Precizați câte dintre următoarele expresii au valoarea **1** (pentru limbajul C/C++), respectiv **true** (pentru limbajul Pascal) dacă și numai dacă $x \in (-\infty, -10) \cup [10, 100)$.

Limbajul C/C++

1. **x< -10 && x>=10 && x<100**
2. **x< -10 || x>=10 && x<100**
3. **!(x>= -10) || !(x>=10) || x<100**

Limbajul Pascal

1. **(x< -10) and (x>=10) and (x<100)**
2. **(x< -10) or ((x>=10) and (x<100))**

- | | |
|--|--|
| 4. <code>x < -10 !(x >= 10) x >= 100</code> | 3. <code>not(x >= -10) or not(not(x >= 10) or (x < 100))</code> |
| 5. <code>x < -10 && x >= 10</code> | 4. <code>(x < -10) or not(not(x >= 10) or (x >= 100))</code> |
| | 5. <code>(x < -10) and (x >= 10)</code> |
| a) 4 b) 1 c) 3 | d) 0 e) 5 f) 2 |

8. În secvența de mai jos, variabilele **i**, **j**, **x** și **y** sunt de tip întreg, iar variabila **a** memorează un tablou bidimensional în care prima linie și prima coloană sunt numerotate cu 1. Toate elementele tabloului primesc valori în urma executării secvenței. Precizați care este valoarea elementului **a[4][4]** (pentru limbajul C/C++), respectiv **a[4,4]** (pentru limbajul Pascal) în urma executării secvenței de mai jos.

Limbajul C/C++

```
x=7928;
for (j=1; j<=4; j++)
{
    y=x;
    for(i=1; i<=4; i++)
    {if (j%2==0)
        a[i][j]=10-y%10;
      else
        a[i][j]=y%10;
      y=y/10; }
    x++;
}
```

- a) 7 b) 9 c) 2

Limbajul Pascal

```
x:=7928;
for j:=1 to 4 do
begin
    y:=x;
    for i:=1 to 4 do
    begin
        if j mod 2=0 then
            a[i,j]:=10-y mod 10
        else
            a[i,j]:=y mod 10;
        y:=y div 10; end;
    x:=x+1;end;
```

- d) 8 e) 3 f) 1

9. Precizați numărul maxim de componente conexe pe care le poate avea un graf neorientat cu 30 de noduri și 20 muchii.

- a) 25 b) 15 c) 10 d) 24 e) 5 f) 30

10. Se consideră un graf neorientat cu 9 noduri, al cărui vector de muchii este **M = {(1,2), (1,9), (2,3), (3,4), (3,7), (3,8), (4,5), (5,6), (5,7), (6,7), (6,8), (8,9)}**. Indicați numărul minim de muchii care trebuie eliminate astfel încât graful să devină eulerian, dar să nu mai fie hamiltonian.

- a) 2 b) 0 c) 4 d) 3 e) 1 f) 5

11. Se utilizează metoda backtracking pentru a genera șiruri de câte 5 caractere din mulțimea {**a, 1, b, 2, c, 3, d, 4**} cu proprietatea că nu poate să aibă două cifre sau două litere alăturate. Știind că primul șir generat este **a1a1a**, iar al doilea este **a1a1b**, indicați șirul obținut imediat înainte de **2c1a1**.

- a) **2b1a1** b) **2b4d3** c) **2b4d4** d) **2c4d4** e) **1c4d4** f) **3c4d4**

12. Precizați valoarea variabilei **a** la finalul executării secvenței următoare de program.

Limbajul C/C++

Limbajul Pascal

<pre>char a[50]="dicarapetacul"; int i=0,j=strlen(a)-1; while(i<=j) { if(a[i]==a[j]) { a[i]=a[i+1]; a[j]=a[j]+1; } i++; j--; } }</pre>	<pre>var a:string[50]; i,j:integer; a:='dicarapetacul';i:=1; j:=length(a); while i<=j do begin if a[i]=a[j] then begin a[i]:=a[i+1]; a[j]:=CHR(ORD(a[j])+1); end; i:=i+1; j:=j-1; end; end;</pre>
a) diarraeetbdul b) diarrafetbdul c) diarraeetuuul d) diarraeetuuul e) diaraeetuuul f) diarraeetuuul	

13. În secvența de mai jos, variabilele **i**, și **j** sunt de tip întreg, iar variabila **a** memorează un tablou unidimensional în care primul element este numerotat cu **1**. Toate elementele tabloului primesc valori în urma executării secvenței. Precizați care este valoarea elementului **a[4]** în urma executării secvenței de mai jos.

Limbajul C/C++ <pre>for(i=1;i<=5;i++) a[i]=10-i; for(i=1;i<=5;i++) if(i<3) a[i]=a[i]+a[6-i]; else a[i]=a[i]-a[6-i];</pre>	Limbajul Pascal <pre>for i:=1 to 5 do a[i]:=10-i; for i:=1 to 5 do if i<3 then a[i]:=a[i]+a[6-i] else a[i]:=a[i]-a[6-i];</pre>
a) 2 b) 4 c) 0	d) 8 e) -9 f) -8

14. Indicați valorile variabilelor **a** și **b**, în urma apelului **f(a,b)** (pentru **Limbajul C++/Pascal**), respectiv **f(&a,b)** (pentru **Limbajul C**), al subprogramului **f** definit mai jos.

Limbajul C++ <pre>int a,b; void f(int&x,int y) { int b,c; a++; b++; x=x*2; y=y*3; b=x; c=y; c++; }</pre> Limbajul C <pre>int a,b; void f(int*x,int y) { int b,c;</pre>	Limbajul Pascal <pre>var a,b:integer; procedure f(var x:integer; y:integer); var b,c:integer; begin inc(a); inc(b); x:=x*2; y:=y*3; b:=x; c:=y; inc(c); end;</pre>
--	--


```

    a++; b++; *x=*x*2;
y=y*3; b=*x; c=y; c++; }

```

- a) Eroare de compilare b) 2 0 c) 2 2 d) 2 3 e) 1 1 f) 1 0

15. Se consideră subprogramul **f** definit mai jos. Precizați ce valoare va avea **f(20,2)**.

Limbajul C++/C

```

int f(int x,int y)
{int p=1;
  if(x>1)
  { while(x%y==0)
    { p=p*y; x=x/y; }
  if(p!=1)
return p+f(x,y+1);
    else return f(x,y+1);
  }
  else return 0;
}

```

- a) 9 b) 4 c) 10

Limbajul Pascal

```

function
f(x,y:integer):integer;
var p:integer;
begin
  p:=1;
  if x>1 then
  begin
    while x mod y=0 do
    begin
      p:=p*y; x:=x div y; end;
    if p<>1 then f:=p+f(x,y+1)
      else f:=f(x,y+1);
    end
  else f:=0; end;
end

```

- d) 7 e) 11 f) 5

Varianta 16

1. Se consideră variabilele de tip întreg $a=15$, $b=30$, $c=5$, $d=10$ și R , indicați valoarea variabilei R în urma executării instrucțiunii:

Limbajul C/C++ $R=a+b/c+d$; **Limbajul Pascal** $R:=a + b \text{ div } c + d$;

a) 19 b) 17 c) 3 d) 31 e) 20 f) 15

2. Fie următoarele două secvențe de cod:

Limbajul C/C++

Secvența 1:

```
s=0;
for (i=1; i<=n; i++)
    s=s+i*i;
```

Secvența 2:

```
s=0; i=<initial>;
do
{ s=s+i*i;
<instrucțiune>
} while (i>=1);
```

Indicați cu ce trebuie înlocuite **<initial>** și **<instrucțiune>** astfel încât cele două secvențe de cod să fie echivalente (în final variabila s să aibă aceeași valoare).

Limbajul C/C++

a) n și $i=i-1$;
b) 1 și $i=i+1$;
c) n și $i=i+1$;
d) 0 și $i=i+1$;
e) $n+1$ și $i=i+1$;
f) 0 și $i=i-1$;

Limbajul Pascal

Secvența 1:

```
s:=0;
for i:=1 to n do
    s:=s+i*i;
```

Secvența 2:

```
s:=0; i:=<initial>;
repeat
s:=s+i*i;
<instrucțiune>
until i=0;
```

Limbajul Pascal

a) n și $i:=i-1$;
b) 1 și $i:=i+1$;
c) n și $i:=i+1$;
d) 0 și $i:=i+1$;
e) $n+1$ și $i:=i+1$;
f) 0 și $i:=i-1$;

3. Precizați secvența de instrucțiuni echivalentă cu următoarea secvență de cod.

Limbajul C/C++

```
if (a>b)
    if (a%2==0)
        if (b%2==0) c=a;
    else c=b;
```

Limbajul C/C++

a) $\text{if} (a>b \ \&\& \ a\%2==0 \ \&\& \ b\%2==0) \ c=a$;
 $\text{if} (a>b \ \&\& \ a\%2==0 \ \&\& \ b\%2!=0) \ c=b$;
b) $\text{if} (a>b \ \&\& \ a\%2==0 \ \&\& \ b\%2==0) \ c=a$;

Limbajul Pascal

```
if a>b then
    if a mod 2=0 then
        if b mod 2=0 then c:=a
    else c:=b;
```

Limbajul Pascal

a) $\text{if} (a>b) \text{ and } (a \bmod 2=0) \text{ and } (b \bmod 2=0) \text{ then } c:=a$;
 $\text{if} (a>b) \text{ and } (a \bmod 2=0) \text{ and } (b \bmod 2<>0) \text{ then } c:=b$;
b) $\text{if} (a>b) \text{ and } (a \bmod 2=0) \text{ and } (b \bmod 2=0) \text{ then } c:=a$
 $\text{else } c:=b$;

```

else c=b;
c) if (a>b && a%2==0 &&
    b%2==0) c=a;
    if (a<b && a%2!=0 &&
        b%2!=0) c=b;

d) if (a>b && a%2==0 &&
    b%2==0) c=a;
    if (a<=b && a%2!=0 &&
        b%2!=0) c=b;

e) if (a>b && a%2==0 &&
    b%2==0) c=a;

f) if (a<=b && a%2!=0 &&
    b%2!=0) c=b;

```

```

c) if (a>b) and (a mod 2=0)
    and (b mod 2=0) then c:=a;
    if (a<b) and (a mod 2<>0)
    and (b mod 2<>0) then c:=b;

d) if (a>b) and (a mod 2=0)
    and (b mod 2=0) then c:=a;
    if (a<=b) and (a mod 2<>0)
    and (b mod 2<>0) then c:=b;

e) if (a>b) and (a mod 2=0)
    and (b mod 2=0) then c:=a;

f) if (a<=b) and (a mod 2<>0)
    and (b mod 2<>0) then c:=b;

```

4. Indicați numărul minim de muchii ce trebuie eliminate dintr-un graf neorientat complet care are **88** noduri astfel încât acesta să devină eulerian.

a) 0 b) 84 c) 88 d) 176 e) 10 f) 44

5. Precizați descendenții nodului **5** din arborele dat de următorul vector tați **(5, 8, 4, 0, 4, 5, 3, 6, 7, 8)**.

a) 1, 2, 6, 8, 10 b) 1 și 6 c) 4 d) $\frac{1, 6}{7}$ e) 1 și 7 f) 6 și 7

6. Se consideră un tablou bidimensional **A**, cu **n** linii și **n** coloane, notăm cu **A_{i,j}** elementul aflat pe linia **i** și coloana **j** ($1 \leq i \leq n, 1 \leq j \leq n$). Precizați condiția necesară ca elementul **A_{i,j}** să fie situat pe prima diagonală de sub diagonală secundară, care este paralelă cu aceasta.

a) $i+j==n-1$ (C/C++) b) $i+j==n+1$ (C/C++) c) $i+j==n+2$ (C/C++)
 $i+j=n-1$ (Pascal) $i+j=n+1$ (Pascal) $i+j=n+2$ (Pascal)
d) $i==j+1$ (C/C++) e) $i==j$ (C/C++) f) $i+j==n-2$ (C/C++)
 $i=j+1$ (Pascal) $i=j$ (Pascal) $i+j=n-2$ (Pascal)

7. Precizați intervalul căruia îi aparține valoarea memorată de variabila reală **x**, astfel încât expresia următoare să aibă valoarea **1** (pentru **Limbajul C/C++**), **true** (pentru **Limbajul Pascal**)

Limbajul C/C++
 $x < -10 \ || \ !(x \geq 10) \ || \ x \geq 100$

Limbajul Pascal
 $(x < -10) \ or \ not(not(x \geq 10) \ or \ (x \geq 100))$

a) $x \in (-\infty, -10) \cup [10, 100)$
b) $x \in (-\infty, -10] \cup [10, 100)$

- c) $x \in (-\infty, -10) \cup (10, 100)$
- d) $x \in (-\infty, -10) \cup [10, 100]$
- e) $x \in (-10, 10) \cup (100, +\infty)$
- f) $x \in (-10, 100)$

8. În secvența de mai jos, variabilele **i**, **j**, **x** și **y** sunt de tip întreg, iar variabila **a** memorează un tablou bidimensional în care prima linie și prima coloană sunt numerotate cu **1**. Toate elementele tabloului primesc valori în urma executării secvenței. Precizați care este valoarea elementului **a[4][4]** (pentru **limbajul C/C++**), respectiv **a[4,4]** (pentru **limbajul Pascal**) în urma executării secvenței de mai jos.

Limbajul C/C++

```
x = 3478;
for (j=4; j>=1; j--)
{
    y=x;
    for (i=4; i>=1; i--)
    {
        if (j%2==0) a[i][j]=10-y%10;
        else a[i][j]=y%10;
        y=y/10;
    }
    x++;
}
```

Limbajul Pascal

```
x:=3478;
for j:=4 downto 1 do
begin
    y:=x;
    for i:=4 downto 1 do
    begin
        if j mod 2=0 then
            a[i,j]:=10-y mod 10
        else a[i,j]:=y mod 10;
        y:=y div 10; end;
    end;
    x:=x+1; end;
```

- a) 7 b) 3 c) 6 d) 4 e) 2 f) 8

9. Precizați numărul maxim de noduri izolate pe care le poate avea un graf neorientat cu **30** de noduri și **20** de muchii

- a) 24 b) 23 c) 15 d) 25 e) 0 f) 10

10. Se consideră un graf neorientat cu **9** noduri, al cărui vector de muchii este **M = {(1,2), (1,9), (2,3), (3,4), (3,7), (3,8), (4,5), (5,6), (5,7), (6,7), (6,8), (8,9)}**. Indicați numărul minim de muchii care trebuie eliminate astfel încât graful să devină eulerian, dar să rămână hamiltonian.

- a) 4 b) 0 c) Nu se poate d) 2 e) 7 f) 3

11. Se utilizează metoda backtracking pentru a genera șiruri de câte **5** caractere distincte din mulțimea **{a, 1, b, 2, c, 3, d, 4}** cu proprietatea că nu poate să aibă două cifre sau două litere alăturate. Știind că primul șir generat este **a1b2c**, iar al doilea este **a1b2d**, precizați șirul obținut imediat înainte de **2c4a1**.

- a) 2c3d4 b) 2c1b4 c) 2b4d3 d) 2c3a4 e) 1c3a4 f) 2c1a4

12. Indicați valoarea variabilei **a** la finalul executării secvenței următoare de program.

Limbajul C/C++

```
char a[]="15iunie1970";
char v[]="aeiouAEIOUn";
int i;
for (i=0; i<strlen(a); i++)
```

Limbajul Pascal

```
var a,v:string;
    i:integer;
...
a:='15iunie1970';
```

<pre>if(a[i]>='0' && a[i]<='9') a[i]=v[a[i]-'0'];</pre>	<pre>v:='aeiouAEIOUn'; for i:=1 to length(a) do if (a[i]>='0') and (a[i]<='9') then a[i]:=v[ORD(a[i])-ORD('0')+1];</pre>
---	--

- | | | |
|------------------------|----------------|-----------------|
| a) Eroare de compilare | b) iEiunieinOe | c) eAiunieeeUIa |
| d) 152410211970 | e) EiunieinO | f) AiunieeeUI |

13. Se consideră un tablou unidimensional **a** cu **n** numere naturale. Dacă pentru **n** se citește valoarea 7, iar a primește valorile: 7, 4, 8, 2, 9, 6 și 2, precizați ce se va afișa la sfârșitul executării secvenței următoare de program.

Limbaajul C++

```
int a[15],i,n,j;
cin>>n;
for(i=0;i<n;i++)
    cin>>a[i];
for(i=0;i<n;i++)
    if(a[i]%2==0)
    {
        for(j=i;j<n-1;j++)
            a[j]=a[j+1];
        n--;
    }
for(i=0;i<n;i++)
    cout<<a[i]<<" ";
```

Limbaajul Pascal

```
var a:array[1..20] of
integer;
    i,j,n,k:integer;
.....
read(n);
for i:=1 to n do read(a[i]);
i:=1;
while i<=n do
begin
    if a[i] mod 2=0 then
begin
        k:=n-1;
        for j:=i to k do
a[j]:=a[j+1];
        n:=n-1; end;
        i:=i+1; end;
for i:=1 to n do
write(a[i], ' ');
```

Limbaajul C

```
int a[15],i,n,j;
scanf("%d",&n);
for(i=0;i<n;i++)
    scanf("%d",&a[i]);
for(i=0;i<n;i++)
    if(a[i]%2==0)
    {
        for(j=i;j<n-1;j++)
            a[j]=a[j+1];
        n--;
    }
for(i=0;i<n;i++)
    printf("%d ",a[i]);
```

- | | | | | | |
|---------|-------|---------|----------|--------|---------|
| a) 7292 | b) 79 | c) 7892 | d) 74296 | e) 426 | f) 4826 |
|---------|-------|---------|----------|--------|---------|

14. Indicați valorile variabilelor **a** și **b**, în urma apelului **f(a,b,b)** (pentru **Limbajul C++/Pascal**), respectiv **f(&a,b,b)** (pentru **Limbajul C**), al subprogramului **f** definit mai jos.

Limbajul C++

```
int a,b;
void f(int&x,int y,int b)
{ a++; b++;
  x=x*2; y=y*3;
}
```

Limbajul C

```
int a,b;
void f(int*x,int y,int b)
{ a++; b++;
  *x=*x*2; y=y*3;
}
```

Limbajul Pascal

```
var a,b:integer;
procedure f(var x:integer;
y,b:integer);
begin
  inc(a); inc(b);
  x:=x*2; y:=y*3;
end;
```

- a) Eroare de compilare b) 2 3 c) 1 1 d) 2 0 e) 2 2 f) 1 0

15. Se consideră subprogramul **f** definit mai jos. Precizați ce valoare va avea **f(1234,6789,1)**.

Limbajul C++/C

```
int f(int x,int y, int p)
{
  if (y!=0)
    if (y%2==0)
      return
        y%10*p+f(x/10,y/10,p*10);
    else
      return
        x%10*p+f(x/10,y/10,p*10);
    else return 0;
}
```

Limbajul Pascal

```
function
f(x,y,p:integer):integer;
begin
  if y<>0 then
    if y mod 2=0 then
      f:= y mod 10 * p +
        f(x div 10,y div 10,p*10)
    else f:=x mod 10 * p +
        f(x div 10,y div 10,p*10)
    else f:=0;
end;
```

- a) 1739 b) 4862 c) 6789 d) 200 e) 6284 f) 1234

Varianta 17

1. Alegeți secvențele de instrucțiuni prin care variabilei întregi **pc** i se atribuie valoarea primei cifre a unui număr natural dat prin variabila **a**.

Limbaajul C++

1. `pc=a/10;`
2. `pc=a;while (pc>9)pc=pc/10;`
3. `pc=a%10;`
4. `pc=a;while (pc>9)pc=pc%10;`
5. `pc=a;while (pc>0)pc=pc/10;`
6. `pc=a;while (pc>0)pc=pc%10;`

Limbaajul C

1. `pc=a/10;`
2. `pc=a;while (pc>9)pc=pc/10;`
3. `pc=a%10;`
4. `pc=a;while (pc>9)pc=pc%10;`
5. `pc=a;while (pc>0)pc=pc%10;`
6. `pc=a;while (pc>0)pc=pc/10;`

a) 1

b) 2

c) 3

Limbaajul Pascal

1. `pc:=a div 10;`
2. `pc:=a; while (pc>9) do pc:=pc div 10;`
3. `pc:=a mod 10;`
4. `pc:=a; while (pc>9) do pc:=pc mod 10;`
5. `pc:=a; while (pc>0) do pc:=pc div 10;`
6. `pc:=a; while (pc>0) do pc:=pc mod 10;`

d) 4

e) 5

f) 6

2. Precizați care dintre următoarele secvențe de instrucțiuni atribuie variabilei întregi **p**, valoarea 3^n , unde variabila **n** reprezintă un număr natural dat.

Limbaajul C++

1. `p=1;for (i=1;i<=n;i++)p*=n;`
2. `p=3;for (i=1;i<=n;i++)p*=i;`
3. `p=1;i=0;while (i<n)p*=3;i++;`
4. `p=1;for (i=1;i<=n;i++)p*=3;`
5. `p=3;for (i=n;i>=0;i--)p*=3;`
6. `p=1;for (i=n;i>=0;i--)p*=3;`

Limbaajul C

1. `p=1;for (i=1;i<=n;i++) p*=n;`
2. `p=3;for (i=1;i<=n;i++) p*=i;`
3. `p=1;i=0;while (i<n) p*=3;i++;`
4. `p=1;for (i=1;i<=n;i++) p*=3;`
5. `p=3;for (i=n;i>=0;i--) p*=3;`
6. `p=1;for (i=n;i>=0;i--) p*=3;`

a) 1

b) 2

c) 3

Limbaajul Pascal

1. `p:=1;for i:=1 to n do p:=p*n;`
2. `p:=3;for i:=1 to n do p:=p*i;`
3. `p:=1; i:=0; while (i<n) do p:=p*3; inc(i);`
4. `p:=1;for i:=1 to n do p:=p*3;`
5. `p:=3; for i:=n downto 0 do p:=p*3;`
6. `p:=1; for i:=n downto 0 do p:=p*3;`

d) 4

e) 5

f) 6

3. Indicați valoarea returnată de funcția definită mai jos, la apelul **sp (3)**.

Limbajul C

```
int sp(int n)
{
    if(n>1)
        return
    sp(n-1)+n*(n+1)*(n+2);
    else return 6;}

```

Limbajul C++

```
int sp(int n)
{if(n>1) return
sp(n-1)+n*(n+1)*(n+2);
else return 6;}

```

a) 30

b) 60

c) 90

Limbajul Pascal

```
function
sp(n:integer):integer;
Begin
    if(n>1) then
        sp:=sp(n-1)+n*(n+1)*(n+2)
        else
            sp:=6;
End;

```

d) 120

e) 150

f) 180

4. Indicați expresia corectă din punct de vedere sintactic și care are ca valoare 3^{2020} .

Limbajul C++

1. `exp(2020*log(3))`
2. `exp(log(2020))*exp(3)`
3. `log(3)*exp(2020)`
4. `log(3*exp(2020))`
5. `exp(log(2020)*log(3))`
6. `log(exp(2020*log(3)))`

Limbajul C

1. `exp(2020*log(3))`
2. `exp(log(2020))*exp(3)`
3. `log(3)*exp(2020)`
4. `log(3*exp(2020))`
5. `exp(log(2020)*log(3))`
6. `log(exp(2020*log(3)))`

a) 1

b) 2

c) 3

Limbajul Pascal

1. `exp(2020*ln(3))`
2. `exp(ln(2020))*exp(3)`
3. `ln(3)*exp(2020)`
4. `ln(3*exp(2020))`
5. `exp(log(2020)*ln(3))`
6. `log(exp(2020*ln(3)))`

d) 4

e) 5

f) 6

5. Fie graful neorientat $G=(V,M)$, unde V este mulțimea nodurilor grafului, $\text{card}(V)=n$, respectiv, M este mulțimea muchiilor grafului, iar $\text{card}(M)=m$. Graful dat are p componente conexe. Dacă $n=1010$, $m=2020$ iar $p=100$, precizați care este numărul maximal de cicluri independente care pot fi construite concomitent pe graf. Prin cicluri independente se înțelege, cicluri care conțin cel puțin câte o muchie care aparține doar unuia din ele.

a) 1001

b) 1010

c) 1011

d) 1100

e) 1101

f) 1110

6. Indicați valoarea variabilei `text` după executarea instrucțiunilor de mai jos.

Limbajul C++

```
char text[250];
strncpy(text,
  strstr("Admitere Poli 2020",
    "oli"),9);
text[9]='\0';
```

Limbajul C

```
char text[250];
strncpy(text,
  strstr("Admitere Poli 2020",
    "oli"),9);
text[9]='\0';
```

Limbajul Pascal

```
var
  text:string[250];p:integer;
Begin
  text:='Admitere Poli 2020';
  p:=pos('oli',text);
  text:=copy(text,p,9);
End.
```

- | | | |
|--------------|------------------|------------------|
| a) Admitere | b) Admitere 2020 | c) Admitere Poli |
| d) Poli 2020 | e) oli 2020 | f) 2020 |

7. Indicați valoarea variabilei **text** după executarea secvenței de instrucțiuni de mai jos.

Limbajul C++

```
char text[250];
strcpy(text,strstr("Admitere Politehnica Bucuresti
2020", "Poli")+strlen("240820201731"));
cout<<strcat(text," ADMIS");
```

Limbajul C

```
char text[250];
strcpy(text,strstr("Admitere Politehnica Bucuresti
2020", "Poli")+strlen("240820201731"));
printf(" %s \n", strcat(text," ADMIS"));
```

Limbajul Pascal

```
var text:string[250];p:integer;
Begin
  text:='Admitere Politehnica Bucuresti 2020';
  p:=pos('Poli',text);
  text:=copy(text,p+length('240820201731'),length(text));
  text:=concat(text,' ADMIS'); writeln(text); End.
```

- | | | |
|-------------------------|----------------------|---------------------------|
| a) Admitere 2020 ADMIS | b) Admitere 2020 | c) Bucuresti ADMIS |
| d) Bucuresti 2020 ADMIS | e) Politehnica ADMIS | f) Politehnica 2020 ADMIS |

8. Utilizând metoda backtracking, se generează toate modalitățile de a se îmbrăca un inginer. Știind că el are la dispoziție **12** cămăși, **8** pantaloni și **9** cravate, indicați numărul de modalități de a se îmbrăca, folosind toate cele trei elemente vestimentare, pe care le are inginerul.

- | | | | | | |
|--------|--------|--------|--------|-------|-------|
| a) 864 | b) 204 | c) 168 | d) 108 | e) 96 | f) 29 |
|--------|--------|--------|--------|-------|-------|

9. Precizați care este cea mai mare valoare pe care o poate lua variabila întreagă **n** astfel încât să se afișeze mesajul **Corect**.

Limbaajul C
if (**n**<17-3***n**)
 printf ("Corect");
else
 printf ("Incorect");
a) 17 b) 15

Limbaajul C++
if (**n**<17-3***n**)
 cout<<"Corect";
else
 cout<<"Incorect";
c) 12 d) 10

Limbaajul Pascal
if **n**<17-3***n** **then**
 write ("Corect")
else
 write ("Incorect");
e) 7 f) 4

10. Precizați care este valoarea inițială a variabilei întregi **n** pentru ca secvența de program de mai jos să afișeze \$\$\$\$.

Limbaajul C
while (**n**!=3)
 { **n**--;
 printf ("\$\$"); }
a) 1 b) 2

Limbaajul C++
while (**n**!=3)
 { **n**--;
 cout<<"\$\$"; }
c) 3 d) 3

Limbaajul Pascal
while **n**<>3 **do**
 begin
 dec (**n**) ;
 write ('\$\$') ;**end**;
e) 5 f) nici o valoare

11. Se construiește un tablou bidimensional cu **n** linii și **n** coloane, în variabila **A** prin secvența de mai jos, unde variabila **n** este un număr natural nenul dat.

Limbaajul C
for (**i**=1;**i**<=**n**;**i**++)
for (**j**=1;**j**<=**n**;**j**++)
A[**i**][**j**]=(2***i**+**j**)/2;

Limbaajul C++
for (**i**=1;**i**<=**n**;**i**++)
for (**j**=1;**j**<=**n**;**j**++)
A[**i**][**j**]=(2***i**+**j**)/2;

Limbaajul Pascal
for **i**:=1 **to** **n** **do**
for **i**:=1 **to** **n** **do**
A[**i**,**j**]:=(2***i**+**j**)/2;

Precizați suma elementelor alfate pe diagonala principală a tabloului bidimensional **A**, în urma execuției secvenței de mai sus.

- a) $\frac{3}{4} \cdot (n+1) \cdot n$ b) $\frac{3}{4} \cdot n^2$ c) $\frac{4}{3} \cdot n^2$ d) $\frac{1}{4} \cdot (n+1) \cdot n$ e) $\frac{1}{4} \cdot n^2$ f) $\frac{1}{3} \cdot n^2$

12. Se consideră declarațiile de mai jos. Indicați tipul expresiei **aa.a.a**

Limbaajul C
struct **S1**
{**int** **a**;
char **b**;};
struct **S2**
{**float** **a**; **double**
b;};
struct **S3**
{**struct** **S1** **a**;
struct **S2** **b**;
} **aa**, **bb**;

Limbaajul C++
struct **S1**
{**int** **a**;
char **b**;};
struct **S2**
{**float** **a**; **double**
b;};
struct **S3**
{**struct** **S1** **a**;
struct **S2** **b**;
} **aa**, **bb**;

Limbaajul Pascal
Type **S1**=**Record**
 a:
 integer;
 b: **char**;
End;
S2=**Record**
 a: **real**;
 b: **real**;
End;
S3=**Record**
 a: **S1**;
 b: **S2**;
End;
var **aa**,**bb**:**S3**;

- a) long/ long/ longint b) float/ float/ real c) int/ int/ integer
d) double/ double/ real e) char/ char/char f) **nu** putem avea în înregistrări diferite, câmpuri cu același nume

13. Stabiliți care este valoarea inițială a variabilei naturale **i** pentru ca secvența de program de mai jos să afișeze valorile 1 2 3 4 5 6 7.

Limbajul C

```
k=1;
for(i=...;i<=2020;i++)
{
    printf("%d ",k);
    k++;
}
```

Limbajul C++

```
k=1;
for(i=...;i<=2020;i++)
{
    cout<<k<<" ";
    k++;
}
```

Limbajul Pascal

```
k:=1;
for i=... to 2020 do
Begin
    Write(k, ' ');
    inc(k);
end;
```

- a) 7 b) 17 c) 283 d) 314 e) 2013 f) 2014

14. Se consideră o mulțime **A** cu **n** numere naturale. Precizați care este complexitatea temporală pentru a genera toate submulțimile care au proprietatea că suma elementelor fiecărei submulțimi generate este divizibilă cu **n**.

- a) $O(n \cdot \log(n))$ b) $O(n + \log(n))$ c) $O(2^n)$ d) $O(n^3)$ e) $O(n^2)$ f) $O(n)$

15. Precizați complexitatea timp pentru secvența de program de mai jos.

Limbajul C

```
k=0;
for(int a=n;a>=1;a/=2)
    for(int b=a;b>=1;b--)
        k++;
printf("%i \n",k);
```

Limbajul C++

```
k=0;
for(int a=n;a>=1;a/=2)
    for(int b=a;b>=1;b--)
        k++;
cout<<k;
```

Limbajul Pascal

```
k:=0;a:=n;
while(a>=1) do
begin
    b:=a;
    while(b>=1) do
    begin
        k++;
        dec(b);
    end;
    a:=a div 2;
end;
write(k);
```

- a) $O(n \cdot \log(n))$ b) $O(n + \log(n))$ c) $O(2^n)$ d) $O(n^2)$ e) $O(n)$ f) $O(1)$

Varianta 18

1. Precizați care dintre următoarele secvențe de instrucțiuni atribuie variabilei întregi **pc** valoarea 100, unde **a=559020**.

Limbajul C

1. `pc=a/100*a%10;`
2. `pc=a;while (pc>9)pc=pc/10;
pc*=a%100;`
3. `pc=a%100;`
4. `pc=a;while (pc>9)pc:=pc%10;
pc*=a%100;`
5. `pc=a/10;
while (pc>0)pc=pc/10;
pc*=a%100;`
6. `pc=a;while (pc>0)pc=pc/10;
pc*=a%100;`

Limbajul C++

1. `pc=a/100*a%10;`
2. `pc=a;while (pc>9)pc=pc/10;
pc*=a%100;`
3. `pc=a%100;`
4. `pc=a;while (pc>9)pc=pc%10;
pc*=a%100;`
5. `pc=a/10;
while (pc>9)pc=pc/10;
pc*=a%100;`
6. `pc=a;while (pc>0)pc=pc/10;
pc*=a%100;`

a) 1

b) 2

c) 3

Limbajul Pascal

1. `pc:=a div 10 * a mod 10;`
2. `pc:=a;
while (pc>9) do
 pc:=pc div 10;
pc:=pc*(a mod 100);`
3. `pc:=a mod 100;`
4. `pc:=a;
while (pc>9) do
 pc:=pc mod 10;
pc:=pc*(a mod 100);`
5. `pc:=a/10;
while (pc>0) do
 pc:=pc div 10;
pc:=pc*(a mod 100);`
6. `pc:=a;
while (pc>0) do
 pc:=pc div 10;
pc:=pc*(a mod 100);`

d) 4

e) 5

f) 6

2. Precizați care dintre următoarele secvențe de instrucțiuni de mai jos, atribuie variabilei întregi **p** valoarea împărțirii întregi a numărului **a** la numărul **b**, unde variabilele **a** și **b** reprezintă două numere întregi date.

Limbajul C

1. `p=1;s=0;
for (i=1;i<=b;i++)
 p-=b;s++;`
2. `p=a;
for (i=1;i<=b;i++)p-=i;`
3. `p=1;i=0;
while (i<b) p-=a;i++;`
4. `p=0;s=a;`

Limbajul Pascal

1. `p:=1;s:=0;for i:=1 to b
 do p:=p-n;inc(s);`
2. `p:=a;for i:=1 to b do
 p:=p-i;`
3. `p:=1;i:=0;
while i<b do
 p:=p*a;inc(i);`
4. `p:=0;s:=a;while (s>0)`

```

    for (i=1; i<=b&& s>0; i++)
    {p++; s-=b;}
5. p=a;
   for (i=0; i<=b; i++) p-=b;
6. p=1;
   for (i=0; i<=b; i++) p-=b;

```

Limbajul C++

```

1. p=1; s=0;
   for (i=1; i<=b; i++)
   p-=b; s++;
2. p=a;
   for (i=1; i<=b; i++) p-=i;
3. p=1; i=0; while (i<b) p-=a;
   i++;
4. p=0; s=a;
   for (i=1; i<=b&& s>0; i++)
   {p++; s-=b;}
5. p=a;
   for (i=0; i<=b; i++) p-=b;
6. p=1;
   for (i=0; i<=b; i++) p-=b;

```

a) 1

b) 2

c) 3

```

begin inc(p); s:=s-b;
end;

```

```

5. p:=a; for i:=0 to b do
   p:=p-b;

```

```

6. p:=1; for i:=0 to b do
   p:=p-b;

```

d) 4

e) 5

f) 6

3. Precizați care este valoarea returnată de funcția definită mai jos, la apelul **sp (5)**.

Limbajul C++

```

float sp(int n)
{if (n>1)
return sp(n-1)+1./ (n*(n+1));
else return 0.5;}

```

Limbajul C

```

float sp(int n)
{if (n>1)
return sp(n-1)+1./ (n*(n+1));
else return 0.5;}

```

a) 0.83

b) 0.82

c) 0.81

Limbajul Pascal

```

function
sp(n:integer):real;
Begin
if (n>1) then
sp:=sp(n-1)+1/(n*(n+1))
else
sp:=0.5;
end;

```

d) 0.38

e) 0.28

f) 0.18

4. Precizați care dintre următoarele expresii de mai jos reprezintă valoare polinomului $p = \sum_{k=0}^n (n-k) \cdot X^k$, pentru un număr x real pozitiv, iar n este un număr natural, x și n sunt introduse de la tastatură.

Limbajul C

```

1. p=0; for (k=0; k<=n; k++)
   p=p+(n-k)*pow(x,k);
2. p=1; for (k=0; k<=n; k++)

```

Limbajul Pascal

```

1. p:=0; for k:=0 to n do
   p:=p+(n-k)*exp(k*ln(x));
2. p:=1; for k:=0 to n do

```

```

    p=p+(n-k)*pow(x,k);
3. p=1;for(k=0;k<n;k++)
    p=p+(n-k)*pow(x,k);
4. p=0;for(k=0;k<=n+1;k++)
    p=p+(n-k)*pow(x,k);
5. p=1;for(k=0;k<=n+1;k++)
    p=p+(n-k)*pow(x,k);
6. p=0;for(k=0;k<n;k++)
    p=p+(n-k)*pow(x,k);

```

```

    p:=p+(n-k)*exp(k*ln(x));
3. p:=1;for k:=0 to n-1 do
    p:=p+(n-k)*exp(k*ln(x));
4. p:=0;for k:=0 to n+1 do
    p:=p+(n-k)*exp(k*ln(x));
5. p:=1;for k:=0 to n+1 do
    p:=p+(n-k)*exp(k*ln(x));
6. p:=0;for k:=0 to n-1 do
    p:=p+(n-k)*exp(k*ln(x));

```

Limbaajul C++

```

1. p=0;for(k=0;k<=n;k++)p=p+(n-k)*pow(x,k);
2. p=1;for(k=0;k<=n;k++)p=p+(n-k)*pow(x,k);
3. p=1;for(k=0;k<n;k++)p=p+(n-k)*pow(x,k);
4. p=0;for(k=0;k<=n+1;k++)p=p+(n-k)*pow(x,k);
5. p=1;for(k=0;k<=n+1;k++)p=p+(n-k)*pow(x,k);
6. p=0;for(k=0;k<n;k++)p=p+(n-k)*pow(x,k);

```

a) 1 b) 2 c) 3 d) 4 e) 5 f) 6

5. Fie graful neorientat $G=(V,M)$, unde V este mulțimea nodurilor grafului, $\text{card}(V)=n$, respectiv, M este mulțimea muchiilor grafului, iar $\text{card}(M)=m$. Având la dispoziție cele n noduri se pot construi 32768 de grafuri neorientate distincte, precizați valoarea variabilei n .

a) 15 b) 14 c) 13 d) 8 e) 7 f) 6

6. Precizați care este valoare variabilei **text** după executarea instrucțiunilor de mai jos.

Limbaajul C

```

char text[250]="";
strncpy(text,strstr("Admitere Poli 2020","Poli"),9);
text[9]='\0';
for(int k=strlen(text)-1;k>=0;k--) printf("%c",text[k]);

```

Limbaajul C++

```

char text[250];
strncpy(text,strstr("Admitere Poli 2020","Poli"),9);
text[9]='\0';
for(int k=strlen(text)-1;k>=0;k--) cout<<text[k];
cout<<text;

```

Limbaajul Pascal

```

var text:string[250];
c:char;
p,k:integer;
Begin
text:='Admitere Poli 2020'; p:=pos('Poli',text);

```

```

text:=copy(text,p,length(text));
for k:=1 to length(text) div 2 do
begin
  c:=text[k];text[k]:=text[length(text)-k+1];
  text[length(text)-k+1]:=c;
end;
writeln(text);
End.

```

- a) Poli 2020 b) oli 2020 c) Admitere Poli
 d) 0202 iloP eretimdA e) 0202 eretimdA f) 0202 iloP

7. Precizați ce valoare are variabila **text** după executarea instrucțiunii de mai jos.

Limbaajul C

```

char text[250],nou[250];
strcpy(text,strstr("Admitere Politehnica Bucuresti 2020",
"ere")+strlen("2408"));
strcpy(nou,text); strnset(nou,'X',12);
strncat(text,nou,12); printf("%s \n", text);

```

Limbaajul C++

```

char text[250],nou[250];
strcpy(text,strstr("Admitere Politehnica Bucuresti 2020",
"ere")+strlen("2408"));
strcpy(nou,text); strnset(nou,'X',12);
strncat(text,nou,12); cout<<text;

```

Limbaajul Pascal

```

var text:string[250];
c:char;
p,i:integer;
Begin
text:='Admitere Poli 2020'; p:=pos('Poli',text);
text:=copy(text,p,length(text));
for i:=1 to length(text) div 2 do
begin
c:=text[i]; text[i]:=text[length(text)-i+1];
text[length(text)-i+1]:=c;
end;
writeln(text);
End.

```

- a) Bucuresti 2020XXXXXX b) Politehnica Bucuresti 2020XXXXXX
 c) Politehnica 2020XXXXXXXXXX d) Politehnica Bucuresti 2020XXXXXXXXXXXXXX
 e) Bucuresti 2020XXXXXXXXXXXXXX f) Bucuresti 2020XXXXXX

8. Utilizând metoda backtracking, se generează toate modalitățile de a se forma o echipă de ingineri cu 5 membrii. Echipa trebuie să fie mixtă, formată din exact 2 ingineri și restul

ingineri. Știind că instituția are **24** ingineri, iar ingineri de **3** ori mai multe, care este numărul de echipe de ingineri care se pot forma?

a) **283946040** b) **283948060** c) **283946080** d) **16832340** e) **16832380** f) **16460640**

9. Fiind date două tablouri unidimensionale ordonate, fiecare cu **n** valori, se dorește obținerea unui al treilea tablou unidimensional ordonat, care va conține, toate elementele celor două tablouri în ordine descrescătoare. Algoritmul descris, efectuează în medie, **nr** comparații pentru a ordona elementele celor doi vectori. Numărul **nr** reprezintă complexitatea algoritmului de sortare și este:

a) **$O(n^2)$** b) **$O(n^3)$** c) **$O(n)$** d) **$O(n^2+n)$** e) **$O(\log(3) \cdot n)$** f) **$O(\log(2))$**

10. Fie trei tije numerotate cu **1, 2** și **3**. Problema constă în mutarea celor **n** discuri de pe tija **1**, pe tija **2**, prin intermediul tije **3**, cu următoarele restricții: la fiecare mutare se deplasează un singur disc; discurile se mută numai de pe o tijă pe alta; un disc cu diametru mai mare nu poate fi așezat peste un disc cu diametru mai mic. Pentru **n = 1**, mutăm discul pe ultima tije. Pentru **n=2**, se fac mutările **1→3, 1→2, 3→2**. În cazul în care **n>3** problema se complică. Respectând restricțiile date se realizează un algoritm de rezolvare a problemei. Precizați complexitatea algoritmului de rezolvare al problemei prezentate.

a) **$\theta(n \cdot \log_3(n^3))$** b) **$\theta(3^n \cdot \log_3(n))$** c) **$\theta(3^n)$**
d) **$\theta(2^n)$** e) **$\theta(n^2 \cdot \log(n^2))$** f) **$\theta(2^n \cdot \log(2^n))$**

11. Se construiește un tablou bidimensional cu **n×n** elemente, în variabila **A** prin secvența de mai jos, unde variabila **n** este un număr natural nenul dat de la tastatură.

Limbaajul C	Limbaajul C++	Limbaajul Pascal
for (i=1; i<=n; i++)	for (i=1; i<=n; i++)	for i:=1 to n do
for (j=1; j<=n; j++)	for (j=1; j<=n; j++)	for i:=1 to n do
A[i][j]=(3*i+2*j)/2;	A[i][j]=(3*i+2*j)/2;	A[i,j]:=(3*i+2*j)/2
		;

Precizați care este suma elementelor de pe diagonala secundară a tabloului bidimensional **A**, în urma execuției secvenței de mai sus.

a) **$\frac{5}{4} \cdot (n+1) \cdot n$** b) **$\frac{3}{4} \cdot (n+1) \cdot n$** c) **$\frac{5}{4} \cdot n^2$**
d) **$\frac{3}{4} \cdot n^2$** e) **$\frac{3}{4} \cdot (n-1) \cdot n$** f) **$\frac{5}{4} \cdot (n-1) \cdot n$**

12. Se consideră declarațiile de mai jos. Precizați care este tipul expresiei **bb.b.b**.

Limbaajul C	Limbaajul Pascal
struct S1{ int a;char b;};	Type S1=Record
struct S2{ float a;double b;};	a: integer;
struct S3{struct S1 a;	b: char; End;
struct S2 b;} aa, bb;	S2=Record
	a: real;
	b: real;
	End;
	S3=Record
	a: S1;
	b: S2;


```
End;
var aa,bb:S3;
```

- a) long/ long/ longint b) float/ float/ real c) double/double/ real
d) int/ int/ integer e) char/ char/ char f) **nu** putem avea în înregistrări diferite, câmpuri cu același nume

13. Precizați care vor fi valorile afișate în urma rulării programului de mai jos pentru variabilele **a=2020**, iar **b=17**.

Limbajul C

```
#include<stdio.h>
int a,b;
void f(int n,int m)
{
if(n!=m)
if(n>m)
f(n-m,m);
else
f(n,m-n);
else
{printf("%i",a*b/n);
printf("%i ",m);
}
}
int main()
{scanf("%i",&a);
scanf("%i",&b);
f(a,b);
return 0;
}
```

Limbajul C++

```
#include<iostream>
using namespace
std;
int a,b;
void f(int n,int m)
{
if(n!=m)
if(n>m)
f(n-m,m);
else
f(n,m-n);
else
cout<<a*b/n<<"
"<<m;
}
int main()
{ cin>>a>>b;
f(a,b);
return 0;
}
```

Limbajul Pascal

```
var a,b:integer;
procedure
f(n,m:integer);
Begin
if n<>m then
if n>m then
f(n-m,m)
else
f(n,m-n)
else
write(a*b div n, '
',m);
end;
begin
readln(a,b);
f(a,b);
end.
```

- a) 8080 4 b) 6060 5 c) 4040 6 d) 2020 2020 e) 34340 1 f) 16144 2

14. Se consideră un graf neorientat conex cu **n** noduri și **m** muchii, iar gradul fiecărui nod este par. Precizați care este complexitatea temporală pentru determinarea unui ciclu eulerian în acest graf pentru algoritmi care pornesc de la o parcurgere. Graful este reprezentat folosind liste de adiacență.

- a) $O(n+m)$ b) $O(n)$ c) $O(n \cdot m)$ d) $O(m)$ e) $O(m \cdot \log(2))$ f) $O(n \cdot \log(2))$

15. Precizați complexitatea timp pentru secvența de program de mai jos.

Limbajul C

```
k=0;
for(int a=n;a>=1;a--)
for(int b=n;b>=1;b--)
k++;
printf("%i \n",k);
```

Limbajul Pascal

```
k:=0;
a:=n;
while(a>=1) do
begin
b:=n;
```

Limbajul C++

```
k=0;
for(int a=n;a>=1;a--)
    for(int b=n;b>=1;b--)
        k++;
cout<<k;
```

a) $O(n \cdot \log n)$ b) $O(2^n)$ c) $O(n^3)$

```
while(b>=1) do
begin
    k++;
    dec(b);
end;
dec(a);
end;
write(k);
```

d) $O(n^2)$ e) $O(n)$ f) $O(1)$

Varianta 19

1. Ce se va afișa în urma rulării secvenței de cod de mai jos:

Limbajul C++

```
int main() {
    int p, *q; p = 45; q = &p;
    cout << q[0];
}
```

Limbajul C

```
int main() {
    int p,*q; p = 45; q = &p;
    printf("%d", q[0]);
}
```

Limbajul Pascal

```
var p:integer; q:^integer;
begin
    p := 45; q := @p;
    write(q^);
end.
```

- a) 100 b) 45 c) Eroare d) 0 e) 43 f) 1

2. Fie secvența de cod următoare, în care se consideră că variabilele **a**, **i**, **n** rețin numere întregi:

Limbajul C++

```
cin>>n; a = 1; i = 2;
while(i<n && a>0){
    if(n%i == 0) a=0;
    else
        i++; cout<<i;}

```

Limbajul C

```
scanf("%d",&n); a = 1; i = 2;
while (i<n && a>0) {
    if(n%i == 0) a = 0;
    else i++; printf("%d",i);}

```

Limbajul Pascal

```
read(n);
a := 1; i := 2;
while ((i<n) and (a>0)) do
begin
    if(n mod i = 0) then
        a:=0
    else
        inc(i);
        write(i);
end;
```

Definim, în acest context, **operație** drept o instrucțiune de **atribuire** sau o expresie de **incrementare**. Care este numărul **minim** de **operații** ce se pot executa în secvența de mai sus, în funcție de valoarea citită pentru variabila **n**?

- a) $2n-1$ b) $n-1$ c) 5 d) 3 e) 2 f) 4

3. Fie graful orientat $G=(V,U)$ unde $V=\{1,2,3,4,5,6,7\}$ este mulțimea vârfurilor, iar $U=\{(2,1), (2,3), (5,2), (5,6), (3,4), (4,5), (4,7), (6,7)\}$ reprezintă mulțimea arcelor. Câte componente tare conexe conține graful?

- a) 6 b) 2 c) 3 d) 1 e) 5 f) 4

4. Fie arborele cu rădăcină cu nodurile numerotate de la 1 la 15, reprezentat prin vectorul de tați: $\{3, 8, 5, 5, 0, 8, 3, 5, 1, 7, 7, 5, 4, 3, 6\}$. Câți descendenți are nodul 3?

- a) 3 b) 4 c) 5 d) 6 e) 2 f) 7

5. Fie secvența de cod următoare, în care se consideră că variabilele **x** și **y** rețin numere întregi:

Limbajul C++

```
void q (..... , .....)  
{ x = 10; y = 20;}  
int main() { x = 1; y = 2;  
            q(x,y); cout<<x<<y;  
            q(y,x); cout<<x<<y;  
}
```

Limbajul C

```
void q (..... , .....)  
    { *x = 10; y = 20;}  
int main() { x = 1; y = 2;  
    q(&x,y); printf("%d%d",x,y);  
    q(&y,x); printf("%d%d",x,y);  
}
```

Limbajul Pascal

```
procedure q( ... , ... ) ;  
begin  
    x := 10; y := 20;  
end;  
begin  
    x := 1; y := 2;  
    q(x,y); write(x,y);  
    q(y,x); write(x,y);  
end.
```

Care este varianta corectă a parametrilor formali din antetul subprogramului **q** pentru care se va afișa secvența **1021010**?

Limbajul C++

- | | | | | | |
|--------------|----------------|---------------|---------------|---------------|--------------|
| a) | b) | c) | d) | e) | f) |
| int x, int y | int &x, int &y | int &x, int y | int x, int &y | int y, int &x | int y, int x |

Limbajul C

- | | | | | | |
|-------------|----------------|---------------|---------------|---------------|--------------|
| a) | b) | c) | d) | e) | f) |
| int x,int y | int *x, int *y | int *x, int y | int x, int *y | int y, int *x | int y, int x |

Limbajul Pascal

- | | | | | | |
|--------------------------|-----------------------------------|-----------------------------------|------------------------------|------------------------------|--------------------------|
| a) | b) | c) | d) | e) | f) |
| x:integer;
y:integer; | var x:integer;
var y: integer; | var x:
integer;
y: integer; | x:integer;
var y:integer; | y:integer;
var x:integer; | y:integer;
x:integer; |

6. Numărul grafurilor complete orientate cu **24** de noduri este:

- | | | | | | |
|--------------|--------------|--------------|--------------|--------------|--------------|
| a) 2^{276} | b) 9^{138} | c) 3^{138} | d) 4^{256} | e) 9^{276} | f) 2^{256} |
|--------------|--------------|--------------|--------------|--------------|--------------|

7. Fie secvența de cod următoare în care se consideră că variabilele **u** și **v** rețin numere întregi:

Limbajul C++

```
int main() { u = 4; v = 4;  
cout << u++*++v;  
u>v ? cout<<"u" : cout <<"v";}
```

Limbajul Pascal

```
begin  
    u:=4; v:=4; inc(v);  
    write (u*v);  
    if(u>v) then  
        write('u')
```

Limbajul C

```
int main() {u = 4; v = 4;
printf("%d", u++*++v);
u>v ? printf("u") :
printf("v");}
```

Ce se va afișa în urma rulării secvenței:

- a) 20v b) 25v c) 20u d) 25u e) 16u f) 16v

```
else
    write('v');
end.
```

8. Se consideră graful orientat $G=(V,U)$ unde $\text{card}(V) = 6$ și $U=\{(3,1),(1,2),(2,3),(4,1),(2,5),(5,3),(3,4)\}$. Indicați numărul minim de muchii ce trebuie eliminate pentru a deveni aciclic?
- a) 5 b) 2 c) 3 d) 4 e) 0 f) 1
9. Câte grafuri neorientate distincte cu 4 noduri care au adiacente nodurile 1 și 2, respectiv nodurile 3 și 4 sunt? Două grafuri se consideră distincte dacă matricile lor de adiacență sunt diferite.
- a) 18 b) 15 c) 20 d) 12 e) 16 f) 10
10. Fie secvența de cod următoare, unde toate variabilele rețin numere întregi și $n < m$:

Limbajul C++

```
int main()
{ k=0;
  cin>>n>>m;
  for(i=1;i<=n;i++) cin>>v[i];
  for(i=1;i<=m;i++)
  { cin>>x; li=1; ls=n;
    while(li<=ls)
    { m1=(li+ls)/2;
      if(x==v[m1]) {li=ls+1;k++;}
      else
      { if(x>v[m1]) li=m1+1;
        else ls=m1-1;}
    } cout<<k; }
```

Limbajul C

```
int main( ){
  k=0; scanf("%d%d", &n, &m);
  for(i=1;i<=n;i++)
    scanf("%d", &v[i]);
  for(i=1;i<=m;i++)
  { scanf("%d", &x); li=1; ls=n;
    while(li<=ls) { m1=(li+ls)/2;
      if(x==v[m1]) {li=ls+1;k++;}
      else
      { if(x>v[m1]) li=m1+1;
```

Limbajul Pascal

```
begin
  k:=0;
  read(n,m);
  for i:=1 to n do
    read(v[i]);
  for i:=1 to m do
    begin
      read(x);
      li := 1; ls := n;
      while(li <= ls) do
        begin
          m1 := (li+ls) div 2;
          if(x = v[m1]) then
            begin
              li:=ls+1; inc(k);
            end
          else
            if(x>v[m1]) then
              li := m1+1
            else ls := m1-1;
          end;
        end;write(k);
      end.
```

```

    else ls = m1-1;}
}   printf("%d",k);
}

```

Care este complexitatea acestei secvențe de cod?

- a) $O(m \cdot \log(n))$ b) $O(n \cdot \log m)$ c) $O(n \cdot m)$
d) $O(n \cdot m \cdot \log n)$ e) $O(n \cdot n)$ f) $O(m \cdot n)$

11. Fie secvența de cod următoare:

Limbajul C++

```

int f(int a[],int li,int ls)
{ if(li==ls) return a[li];
  else
    return f(a,li,(li+ls)/2) +
    f(a,(li+ls)/2+1, ls);}
int main()
{ int n, a[20],i;
  cin>>n;
  for(i=1;i<=n;i++) cin>>a[i];
  cout<<f(a,1,n);
}

```

Limbajul C

```

int f(int a[],int li,int ls)
{
  if(li == ls) return a[li];
  else
    return f(a,li, (li+ls)/2) +
    f(a,(li+ls)/2+1, ls);
}
int main()
{int n,a[20],i;
 scanf("%d", &n);
 for(i=1; i<=n; i++)
   scanf("%d", &a[i]);
 printf("%d",f(a,1,n));
}

```

Care este complexitatea acestei secvențe de cod?

- a) $O(n \cdot \log n)$ b) $O(\log n)$ c) $O(n)$ d) $O(n^2)$ e) $O(n^2+1)$ f) $O(n^2-1)$

12. Se consideră un arbore în care fiecare nod intern (nod care nu este pe ultimul nivel) are doi descendenți direcți. Dacă arborele are 38 niveluri (rădăcina se află pe nivelul 0) câte noduri are arborele?

- a) 2^{37} b) $4^{19}-1$ c) $2^{38}+1$ d) $2^{37}+1$ e) $2^{33}+1$ f) 2^{39}

13. Ce se va afișa pentru secvența de cod:

Limbajul Pascal

```

type
  vector=array[1..20] of
  integer;
var n,i:integer;a:vector;
function f(var a:vector;
  li,ls:integer):integer;
begin
  if(li=ls) then f:=a[li]
  else
    f:=f(a,li,(li+ls) div 2)+
    f(a,(li+ls) div 2+1, ls ) ;
  end;
begin
  read(n);
  for i:=1 to n do
    read (a[i]) ;
    write(f(a,1,n));
end.

```

Limbajul C++

```

int x,y;
void f(int &y, int x)
    {x++; y=y+x;}
int main() {
x = 4; y = 2; cout<<x<<y<<" ";
f(x,y); cout<<x<<y<<" ";
f(x,x); cout<<x<<y<<" ";
f(y,x); cout<<x<<y<<" ";}

```

Limbajul C

```

int x,y;
void f(int *y,int x)
    {x++;*y=*y+x;}
int main() {
x = 4; y = 2; printf("%d%d ",x,y);
f(&x,y); printf("%d%d ", x,y);
f(&x,x); printf("%d%d ", x,y);
f(&y,x); printf("%d%d ", x,y); }

```

a) 42 72 152 1518

b) 42 215 42 518

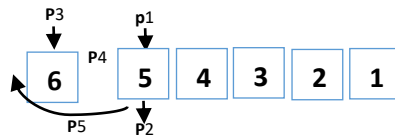
c) 42 27 415 158

d) 42 62 41 58

e) 42 62 45 58

f) 42 72 415 58

14. Folosind algoritmul de sortare prin inserție, pentru ordonarea crescătoare a tabloului unidimensional $v=[6,5,4,3,2,1]$ se efectuează 45 de pași (de exemplu pentru deplasarea elementului cu valoarea 6 pe poziția 2 se execută 5 pași):

P1: $i \leftarrow 2$ P2: $x \leftarrow v[i]$ P3: $j \leftarrow i - 1$;P4: $v[j + 1] \leftarrow v[j]$;P5: $v[j] \leftarrow x$.

Precizați câți pași se execută folosind același algoritm pentru ordonarea crescătoare a tabloului $v=[1000,999,\dots,3,2,1]$

a) 1001999

b) 1001997

c) 1001998

d) 1002000

e) 1001897

f) 1001887

15. Fie subprogramul de mai jos:

Limbajul C++

```

void f(int n,int k) {int i;
for(i = 1; i <= n; i++)
    {if(i%k == 0)cout<<i<<" ";
      f(n-1,k);}}

```

Limbajul C**Limbajul Pascal**

```

procedure f (n,k:integer);
var i : integer;
begin
    for i := 1 to n do
        begin
            if(i mod k = 0) then
                write(i, ' ');

```

<pre>void f(int n,int k){int i; for(i = 1; i <= n; i++) {if(i%k==0)printf("%d ",i); f(n-1,k);}</pre>	<pre> f(n-1,k); end; end;</pre>
---	--

De câte ori se execută instrucțiunea de decizie în cadrul subprogramului, dacă apelul este **f(3,1)**?

- a) **15 ori** b) **14 ori** c) **16 ori** d) **8 ori** e) **9 ori** f) **10 ori**

Varianta 20

1. Fie subprogramul:

Limbajul C/C++

```
int f (int n, int s){
    if (n < s) return 0;
    else
        if(n%s == 0)
            return 1+ f(n/s,s+1);
        else
            return f(n/s,s);
}
```

Limbajul Pascal

```
function f(n,s:integer):
integer;
begin
    if (n < s) then f:=0
    else if (n mod s =0) then
        f:= 1 + f(n div s,s+1)
    else f:= f(n div s,s);
end;
```

Subprogramul se execută pentru următoarele seturi de valori $n=720, s=2$; $n=120, s=3$; $n=120, s=1$; $n=720, s=1$. Pentru câte dintre apeluri subprogramul f va returna valoarea 5?

- a) un apel b) 2 apeluri c) 3 apeluri d) niciun apel e) 4 apeluri f) 5 apeluri

2. Fie subprogramul de mai jos unde n și c sunt variabile întregi:

Limbajul C++

```
int f(int &n, int c){
    int a = n%10;
    if(n == 0) return 0;
    else
        if(a == c)
            {n=n/10;
            return 1+f(n,c);}
    else
        {n=n/10%10;return f(n,c);
        }}
}
```

Limbajul Pascal

```
function f(var n:integer;
c:integer) : integer;
var a : integer;
begin
    a := n mod 10;
    if(n = 0) then f := 0
    else
        if(a = c) then
            begin
                n:=n div 10;
                f:=1+f(n , c);
            end
        else
            begin
                n:=n div 10 mod 10;
                f:=f(n,c);
            end;
    end;
end;
```

Limbajul C

```
int f (int *n, int c) {
    int a = *n %10;
    if(*n == 0) return 0;
    else
        if(a==c)
            {*n=*n/10;
            return 1+f(n,c);}
    else
        {*n=(*n)/10%10;
        return f(n,c);}
}
```

Care sunt variabilele ale căror valori sunt reținute în stiva subprogramului?

- a) n, c, a b) c, a c) n, c d) a e) c f) n, a

3. Fie G un graf neorientat cu $n > 0$ vârfuri și $m > 0$ muchii, reprezentat prin liste de adiacență. Complexitatea unui algoritm care afișează matricea de adiacență asociată grafului este:
a) $O(m \cdot \log n)$ b) $O(m \cdot n)$ c) $O(n^2)$ d) $O(m^2)$ e) $O(m^2+1)$ f) $O(m^2-1)$
4. Variabilele x și y rețin numere întregi. Care dintre expresiile de mai jos are valoarea 1 (Limbajul C/C++, True (Limbajul Pascal) știind că $x > -1$ și $y < 3$?
Limbajul C/C++
a) $x*y+y-3*x-3 > 0$ b) $!(x*y+y-3*x-3 >= 0)$ c) $(x-1) * (y-3) < 0$
d) $(x+1) * (y-3) > 0$ e) $(x-1) * (y+3) > 0$ f) $(x+1) * (y+3) > 0$
Limbajul Pascal
a) $x*y+y-3*x-3 > 0$ b) $NOT(x*y+y-3*x-3 >= 0)$ c) $(x-1) * (y-3) < 0$
d) $(x+1) * (y-3) > 0$ e) $(x-1) * (y+3) > 0$ f) $(x+1) * (y+3) > 0$
5. Se consideră numărul natural $n = 231045$. Dacă se determină toate submulțimile formate din cifrele lui n care au suma valorilor componentelor egală cu 10, câte submulțimi conțin cifra 0?
a) 3 b) 5 c) 2 d) 1 e) 4 f) 6
6. Se consideră șirul $\{a, b, c, u, i, e\}$. Se generează folosind metoda backtracking, în ordine lexicografică, toate cuvintele de trei litere distincte, care conțin două vocale. Dacă primele trei soluții sunt **abe, abi, abu** care este a 9-a soluție?
a) **aic** b) **aib** c) **aec** d) **aub** e) **ace** f) **aei**
7. În câte moduri se poate scrie numărul 12 ca sumă de numere prime?
a) 5 b) 3 c) 7 d) 6 e) 5 f) 4
8. Se consideră mulțimea de cuvinte $\{\text{info, mate, fizica, chimie, biologie}\}$. Se generează folosind metoda backtracking, lexicografic, în ordinea inversă citirii cuvântului, submulțimi de câte trei cuvinte distincte. Dacă primele trei soluții sunt: $\{\text{fizica, biologie, chimie}\}; \{\text{fizica, biologie, mate}\}; \{\text{fizica, biologie, info}\};$ înaintea soluției $\{\text{chimie, mate, info}\}$ este soluția:
a) $\{\text{biologie, mate, info}\}$ b) $\{\text{biologie, chimie, mate}\}$ c) $\{\text{chimie, biologie, info}\}$
d) $\{\text{chimie, mate, biologie}\}$ e) $\{\text{fizica, mate, biologie}\}$ f) $\{\text{chimie, fizica, biologie}\}$
9. Se consideră un arbore cu rădăcină în care fiecare nod intern (nod care nu este pe ultimul nivel) are doi descendenți direcți. Dacă arborele are k niveluri (rădăcina se află pe nivelul 0) câte noduri sunt pe nivelul k ?
a) 2^{k+1} b) $2^{k-1}+1$ c) 2^k d) 2^{k-1} e) $2^{k-2}+1$ f) $2^{k+1}+1$

10. Se consideră șirul primelor $n \times m$ numere naturale unde $n \geq 1$ și $m \geq 1$. Dacă se afișează câte m numere pe o linie, numărul 123 se află pe linia 4 și coloana 3, atunci pe ce linie și coloană se află numărul 167?

a) linia 5, coloana 7 b) linia 4, coloana 7 c) linia 6, coloana 4 d) linia 6, coloana 2 e) linia 5, coloana 2 f) linia 5, coloana 3

11. Fie secvența de cod următoare în care se consideră că variabilele a, i, n rețin numere întregi.

Limbajul C++

```
int main() {
    cin >> n; a = 1; i = 2;
    while (i < n && a > 0) {
        if (n % i == 0) a = 0;
        else i++; cout << i;
    }
}
```

Limbajul C

```
int main() { scanf("%d", &n);
a = 1; i = 2;
while (i < n && a > 0) {
    if (n % i == 0) a = 0;
    else i++;
printf("%d", i);
}
```

Limbajul Pascal

```
read(n); a := 1; i := 2;
while (i < n) and (a > 0) do
begin
    if (n mod i = 0) then
        a := 0
    else
        inc(i);
    write(i);
end;
```

Definim, în acest context, **operație** drept o instrucțiune de **atribuire** sau o expresie de **incrementare**. Care este numărul **maxim** de **operații** ce se pot executa în secvența de mai sus, în funcție de valoarea citită pentru variabila n ?

a) $2n+2$ b) $2n$ c) $n-1$ d) $2n+3$ e) n f) $n+1$

12. Fie secvența de program unde variabila i reține un număr întreg:

Limbajul C++

```
i = 4;
while (i <= 25)
{cout << i/10 + i%10 << " ";
i+=2;
}
```

Limbajul C

```
i = 4;
while (i <= 25)
{printf("%d ", i/10 + i%10);
i+=2;
}
```

Limbajul Pascal

```
i := 4;
while i <= 25 do
begin
    write(i div 10 + i mod 10, ' ');
    i:=i+2;
end;
```

Ultimele trei numere afișate sunt:

a) 2 4 7 b) 2 4 9 c) 6 2 4 d) 9 2 4 e) 2 4 6 f) 9 2 1

13. Fie secvența de cod de mai jos:

Limbajul C/C++

```
float s,p;
float s1(int n){
  if(n==0) return 2;else
  if(n==1) return s;else
  return s*s1(n-1)-
    p*s1(n-2);
}
```

Limbajul Pascal

```
var s,p : float;
function s1(n : integer) :
  real;
begin
  if (n = 0) then s1:=2
  else
  if (n = 1) then s1:=s
  else s1:=s*s1(n-1)-p*s1(n-2);
end;
```

Dacă la apelul subprogramului **s1** se returnează valoarea **82**, ce valori inițiale au variabilele **n**, **s** și **p** în această ordine?

- a) 4 3 4 b) 4 4 3 c) 4 2 3 d) 3 3 2 e) 3 1 2 f) 3 4 2

14. Fie secvența de cod unde toate variabilele sunt întregi:

Limbajul C++

```
s=0; cin>>n>>k;
for(i=1;i<=n;i++)
  for(j=1;j<=n;j++)
  { if (i > j) t = i - j;
    else t = j - i;
    if(i==j || t<=k || j==n-i+1 || (i+j>=n-k+1 && i+j<=n+k+1))
      a[i][j] = 1;
    else a[i][j] = 2;
    if(a[i][j] == 2) s++;}
```

Limbajul C

```
s=0; scanf("%d%d",&n,&k);
for(i=1;i<=n;i++)
  for(j=1;j<=n;j++)
  {if (i > j) t = i - j;
    else t = j - i;
    if(i==j || t<=k || j==n-i+1 || (i+j>=n-k+1 && i+j<=n+k+1))
      a[i][j] = 1;
    else a[i][j] = 2;
    if(a[i][j] == 2) s++;}
```

Limbajul Pascal

```
s:=0; read(n,k);
for i:=1 to n do
  for j:=1 to n do
  begin
    if(i>j) then t:=i-j
    else t:=j-i;
    if((i=j) or (t<=k) or (j=n-i+1) or ((i+j>=n-k+1) and (i+j<=
n+k+1))) then a[i,j]:=1
```

```

else a[i,j]:=2;
if(a[i,j] = 2) then inc(s);
end;

```

Pentru ce valori ale lui n și k variabila s va avea valoarea 8?

a) $n=6; k=2$ b) $n=6; k=1$ c) $n=5; k=2$ d) $n=7; k=1$ e) $n=4; k=2$ f) $n=6; k=3$

15. Se consideră un tablou bidimensional în care $a[i][j]=j+3(i-1)$, ($1 \leq i, j \leq 3$)

Fie secvența de cod de mai jos:

Limbajul C++

```

k = 0;
for(i=1; i<=3; i++)
{for(j =1; j<=3-k; j++)
    cout<<a[α][β]<<" ";
    k++; }

```

Limbajul C

```

k=0;
for(i=1; i<=3; i++)
{for(j =1; j<=3-k; j++)
    printf("%d ",a[α][β]);
    k++;}

```

Limbajul Pascal

```

k := 0;
for i:=1 to 3 do
begin
    for j:=1 to 3-k do
        write(a[α,β], ' ');
    inc(k); end;

```

Ce valori au α și β dacă la execuția secvenței se afișează șirul 3 5 7 2 4 1?

a) $\alpha=3-j-k;$ b) $\alpha=j;$ c) $\alpha=j;$ d) $\alpha=4-j-k;$ e) $\alpha=4+j-k;$ f) $\alpha=4+j+k;$
 $\beta=j$ $\beta=3-j-k$ $\beta=4-j-k$ $\beta=j$ $\beta=j-1$ $\beta=j+1$

Varianta 21

1. Precizați care dintre următoarele expresii are valoarea **true** în Pascal sau **1** în C/C++ dacă și numai dacă numărul întreg **x** are exact trei cifre?

Limbajul C++

- a) `(x%1000==0) || (x%100!=0)`
- b) `(x/10==0) && (x/100==0)`
- c) `(x%10==0) && (x/10==0)`
- d) `(x/1000==0) && (x/100!=0)`
- e) `(x/1000==0) || (x/100==0)`
- f) `!((x/1000==0) && (x/100!=0))`

Limbajul C

- a) `(x%1000==0) || (x%100!=0)`
- b) `(x/10==0) && (x/100==0)`
- c) `(x%10==0) && (x/10==0)`
- d) `(x/1000==0) && (x/100!=0)`
- e) `(x/1000==0) || (x/100==0)`
- f) `!((x/1000==0) && (x/100!=0))`

Limbajul Pascal

- a) `(x mod 1000=0) or (x mod 100<>0)`
- b) `(x div 10=0) and (x div 100=0)`
- c) `(x mod 10=0) and (x div 10=0)`
- d) `(x div 1000=0) and (x div 100<>0)`
- e) `(x div 1000=0) or (x div 100=0)`
- f) `not((x div 1000=0) and (x div 100<>0))`

2. Roboțelul Bob se mișcă într-un plan cartezian. Pentru a reține poziția robotului definim următoarea structură:

Limbajul C

```
typedef struct
{
    float x,y;
} robot;
robot bob;
```

Limbajul C++

```
struct robot
{
    float x,y;
};
robot bob;
```

Limbajul Pascal

```
type robot=record
    x,y:real;
end;
var bob:robot;
```

Precizați care dintre expresiile de mai jos este adevărată dacă și numai dacă roboțelul se află în interiorul sau pe laturile pătratului de coordonate $(-2, -2)$, $(-2, 2)$, $(2, 2)$, $(2, -2)$?

Limbajul C++

- a) `(robot.x>=-2) && (robot.x<=2) && (robot.y>=-2) && (robot.y<=2)`
- b) `(robot.x<=-2) || (robot.x>=2) || (robot.y<=-2) || (robot.y>=2)`
- c) `(bob.x<=-2) || (bob.x>=2) || (bob.y>=-2) || (bob.y<=2)`
- d) `(bob.x>=-2) && (bob.x<=2) && (bob.y>=-2) && (bob.y<=2)`
- e) `(bob.x>=-2) && (bob.x<=2) || (bob.x>=-2) && (bob.x<=2)`
- f) `(robot.x>=-2) && (robot.x<=2) || (robot.x>=-2) && (robot.x<=2)`

Limbajul C

- a) `(robot.x>=-2) && (robot.x<=2) && (robot.y>=-2) && (robot.y<=2)`
- b) `(robot.x<=-2) || (robot.x>=2) || (robot.y<=-2) || (robot.y>=2)`
- c) `(bob.x<=-2) || (bob.x>=2) || (bob.y>=-2) || (bob.y<=2)`
- d) `(bob.x>=-2) && (bob.x<=2) && (bob.y>=-2) && (bob.y<=2)`
- e) `(bob.x>=-2) && (bob.x<=2) || (bob.x>=-2) && (bob.x<=2)`
- f) `(robot.x>=-2) && (robot.x<=2) || (robot.x>=-2) && (robot.x<=2)`

Limbajul Pascal

- a) (robot.x>=-2) and (robot.x<=2) and (robot.y>=-2) and (robot.y<=2)
- b) (robot.x<=-2) or (robot.x>=2) or (robot.y<=-2) or (robot.y>=2)
- c) (bob.x<=-2) or (bob.x>=2) or (bob.y>=-2) or (bob.y<=2)
- d) (bob.x>=-2) and (bob.x<=2) and (bob.y>=-2) and (bob.y<=2)
- e) (bob.x>=-2) and (bob.x<=2) or (bob.x>=-2) and (bob.x<=2)
- f) (robot.x>=-2) and (robot.x<=2) or (robot.x>=-2) and (robot.x<=2)

3. Precizați ce se va afișa în urma execuției următorului program?

Limbajul C++

```
#include <iostream>
using namespace std;
int main()
{   int i,s=0;
    for(i=1;i<=5;i++);
        s=s+i;
    cout<<s; return 0;}
```

Limbajul Pascal

```
var i,s:integer;
begin
s:=0;
for i:=1 to 5 do
    i:=i+1;
s:=s+i; write(s);
end.
```

Limbajul C

```
#include <stdio.h>
int main()
{   int i,s=0;
    for(i=1;i<=5;i++);
        s=s+i;
    printf("%d",s); return 0; }
```

- a) Programul nu va afișa nimic, va genera eroare de compilare. b) 15 c) 6 d) 5 e) 0 f) 10

4. Precizați ce se va afișa pe ecran în urma execuției următorului program?

Limbajul C++

```
#include <iostream>
using namespace std;
int main()
{
    char sir[]="ANA";
    int i=0;
    while(sir[i])
        sir[i++]++;
    cout<<sir;
    return 0;
}
```

Limbajul Pascal

```
var sir:string;
    i:integer;
begin
    sir:='ANA';
    for i:=1 to length(sir) do
        sir[i]:=succ(sir[i]);
    write(sir);
end.
```

Limbajul C

```
#include <stdio.h>
int main()
{
    char sir[]="ANA";
    int i=0;
    while(sir[i])
        sir[i++]++;
    printf("%s",sir);
    return 0;
}
```

- a) ANA b) A c) AN d) BOB e) BAB f) COC

5. Precizați ce se va afișa pe ecran în urma execuției următorului program?

Limbajul C++

```
#include <iostream>
using namespace std;
struct coordonate{
    int abscisa,ordonata;
};
int main()
{
    coordonate abscisa;
    abscisa.abscisa=100;
    abscisa.ordonata=200;
    cout<<abscisa.abscisa<<" ";
    cout<<abscisa.ordonata;
    return 0;}

```

Limbajul Pascal

```
type coordonate=record
    abscisa,ordonata:integer;
end;

var abscisa:coordonate;

begin
    abscisa.abscisa:=100;
    abscisa.ordonata:=200;
    write(abscisa.abscisa,' ');
    write(abscisa.ordonata);
end.
```

Limbajul C

```
#include <stdio.h>
typedef struct {
    int abscisa,ordonata;
}coordonate;
int main()
{
    coordonate abscisa;
    abscisa.abscisa=100;
    abscisa.ordonata=200;
    printf("%d ",abscisa.abscisa);
    printf("%d", abscisa.ordonata);
    return 0;
}
```

- a) Programul nu va afișa nimic, va genera eroare de compilare deoarece numele variabilei de tip struct nu poate coincide cu numele câmpului. b) 0 0 c) 100 200 d) 200 100 e) 100 100 f) 200 200

6. Care va fi valoarea returnată de funcția f prezentată mai jos?

Limbajul C/C++

```
char f()
{ int i,j,mat[5][5];
  char v='a';
  for(i=0;i<5;i++)
    for(j=0;j<5;j++)
      { mat[i][j]=v;
        v++; }
  return mat[2][3]; }
```

Limbajul Pascal

```
type matrice=array[0..4,0..4] of
      char;
function f:char;
  var i,j:integer;
      mat:matrice;
      v:char;
begin
  v:='a';
  for i:=0 to 4 do
    for j:=0 to 4 do
      begin
        mat[i,j]:=v;v:=succ(v);
      end;
  f:=mat[2,3];
end;
```

- a) e b) i c) n d) m e) o f) p

7. Se consideră graful neorientat G reprezentat prin următoarea matrice de adiacență:

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Precizați numărul componentelor conexe ale grafului G.

- a) 1 b) 2 c) 3 d) 4 e) 5 f) 6

8. Corectați secvența de program de mai jos astfel încât să realizeze corect căutarea unui număr întreg **x** într-un vector **v** cu n elemente numere întregi ordonate crescător.

Limbajul C/C++

```
p=0;
u=n-1;
q=0;
while(p<=u && q==0)
{
  m=(p+u)/2;
  if(x==v[m]) q=1;
  else if(x<v[m]) u=m-1;
    else p=m-1;
}
if(q==1)
  printf("Elementul a fost
gasit");
else
```

Limbajul Pascal

```
p:=0;
u:=n-1;
q:=0;
while (p<=u) and (q=0) do
begin
  m:=(p+u) div 2;
  if(x=v[m]) then q:=1
  else if x<v[m] then u:=m-1
    else p:=m-1;
end;
if q=1 then
  write('Elementul a fost
gasit')
else
```

```
printf("Elementul nu a
fost gasit");
```

Limbajul C/C++

- a) Instrucțiunea
`while(p<=u && q==0)`
trebuie înlocuită cu
`while(p>=u && q==0)`
- b) Instrucțiunea `m=(p+u)/2;`
trebuie înlocuită cu `m=(p+u)%2;`
- c) Instrucțiunea `p=m-1;` trebuie
înlocuită cu `p=m+1;`
- d) Instrucțiunea `u=m-1;` trebuie
înlocuită cu `u=m+1;`
- e) Instrucțiunea `if(x<v[m])` trebuie
înlocuită cu `if(x>v[m])`
- f) Instrucțiunea `q=1;` trebuie înlocuită
cu `q=0;`

```
write('Elementul nu a
fost gasit');
```

Limbajul Pascal

- a) Instrucțiunea
`while (p<=u) and (q=0) do`
trebuie înlocuită cu
`while (p>=u) and (q=0) do`
- b) Instrucțiunea
`m:=(p+u) div 2;` trebuie
înlocuită cu `m:=(p+u) mod 2;`
- c) Instrucțiunea `p:=m-1;` trebuie
înlocuită cu `p:=m+1;`
- d) Instrucțiunea `u:=m-1;` trebuie
înlocuită cu `u:=m+1;`
- e) Instrucțiunea `if x<v[m]` trebuie
înlocuită cu `if x>v[m]`
- f) Instrucțiunea `q:=1;` trebuie înlocuită
cu `q:=0;`

9. Precizați ce se va afișa pe ecran în urma execuției programului următor?

Limbajul C++

```
#include <iostream>
using namespace std;
int main()
{ const int m=4,n=5;
  int i,j,aux;
  char a[m][n]=
  { {'a','b','c','d','e'},
    {'f','g','h','i','j'},
    {'k','l','m','n','o'},
    {'p','q','r','s','t'}};
  for(i=0;i<2;i++)
  { aux=a[2][n-1];
    for(j=n-1;j>0;j--)
      a[2][j]=a[2][j-1];
    a[2][0]=aux; }
  for(i=0;i<2;i++)
  { aux=a[m-1][2];
    for(j=m-1;j>0;j--)
      a[j][2]=a[j-1][2];
    a[0][2]=aux; }
  for(i=0;i<m;i++){
    for(j=0;j<n;j++)
      cout<<a[i][j]<<" ";
```

Limbajul Pascal

```
const m=4; n=5;
var i,j:integer;
    aux:char;
    a:array[0..m-1,0..n-1]
      of char=
  (('a','b','c','d','e'),
   ('f','g','h','i','j'),
   ('k','l','m','n','o'),
   ('p','q','r','s','t'));
begin
  for i:=0 to 1 do
  begin
    aux:=a[2,n-1];
    for j:=n-1 downto 1 do
      a[2,j]:=a[2,j-1];
    a[2,0]:=aux; end;
  for i:=0 to 1 do
  begin
    aux:=a[m-1,2];
    for j:=m-1 downto 1 do
      a[j,2]:=a[j-1,2];
    a[0,2]:=aux; end;
  for i:=0 to m-1 do
```

```

    cout<<endl; }
return 0; }

```

```

begin
    for j:=0 to n-1 do
        write(a[i,j], ' ');
    writeln; end;
end.

```

Limbajul C

```

#include <stdio.h>
int main()
{ const int m=4,n=5;
  int i,j,aux;
  char a[4][5]={{'a','b','c','d','e'},
                {'f','g','h','i','j'},
                {'k','l','m','n','o'},
                {'p','q','r','s','t'}};
  for(i=0;i<2;i++)
  { aux=a[2][n-1];
    for(j=n-1;j>0;j--)
      a[2][j]=a[2][j-1];
    a[2][0]=aux;
  }
  for(i=0;i<2;i++)
  {
    aux=a[m-1][2];
    for(j=m-1;j>0;j--)
      a[j][2]=a[j-1][2];
    a[0][2]=aux;
  }
  for(i=0;i<m;i++){
    for(j=0;j<n;j++)
      printf("%c %c",a[i][j], ' ');
    printf("\n");
  }
  return 0;
}

```

a)	b)	c)	d)	e)	f)
a b c d e	a b m d e	d e a b c	a b k d e	e d c b a	a b c d e
f g h i j	f g r i j	i j f g h	f g r i j	f g h i j	f g h i j
n o k l m	k l c n o	l m n o k	n o c l m	k l m n o	k l m n o
p q r s t	p q h s t	s t p q r	p q h s t	p q r s t	t s r q p

10. Precizați ce se va afișa pe ecran în urma execuției următorului program?

Limbajul C++

```

#include <iostream>
using namespace std;
int a=5, b=10, c=15;
void f(int a,int &b,int &c)
{ a=a+5; b=b+10; c=c+15;

```

Limbajul Pascal

```

var a,b,c:integer;
procedure f(a:integer;var
b:integer; var
c:integer);
begin

```

```

}
int main()
{ f(a,b,c);f(a,a,b);
  cout<<"a="<<a;
  cout<<"b="<<b;
  cout<<"c="<<c;
  return 0;
}

```

```

a:=a+5;b:=b+10;c:=c+15;
end;
begin
  a:=5; b:=10; c:=15;
  f(a,b,c);f(a,a,b);
  write('a=',a);
  write('b=',b);
  write('c=',c);
end.

```

Limbaajul C

```

#include <stdio.h>
int a=5, b=10, c=15;
void f(int a, int *b, int *c)
{ a=a+5; *b=*b+10; *c=*c+15; }
int main()
{ f(a,&b,&c); f(a,&a,&b);
  printf("a=%d",a ); printf("b=%d",b); printf("c=%d",c) ;
  return 0; }

```

a) a=5	b) a=15	c) a=10	d) a=5	e) a=15	f) a=10
b=5	b=35	b=20	b=10	b=5	b=5
c=20	c=30	c=30	c=15	c=30	c=30

11. Un echipaj va pleca spre Marte în misiunea POLI. El este alcătuit din căpitan 1 - Andrei, căpitan 2 - Marian și cercetătorii Alina, Dana și Marius. Săptămânal membrii echipajului trebuie să transmită un raport respectând o anumită ordine: întotdeauna raportul căpitanului 1 trebuie să fie înaintea raportului căpitanului 2. Știind că primele trei soluții posibile de raportare sunt:

Andrei Marian Alina Dana Marius

Andrei Marian Alina Marius Dana

Andrei Marian Dana Alina Marius

afișați a zecea soluție.

a) Marius	b) Dana	c) Marius	d) Andrei	e) Andrei	f) Andrei
Dana	Marius	Andrei	Alina	Alina	Alina
Alina	Alina	Marian	Dana	Dana	Marius
Andrei	Andrei	Alina	Marius	Marian	Marian
Marian	Marian	Dana	Marian	Marius	Dana

12. Înlocuiți valoarea lui **v[3]** cu una dintre următoarele valori astfel încât funcția să returneze **0** în C/C++ sau **false** în Pascal pentru apelul **f(5)**.

Limbaajul C++

```

int v[]={15,12,7,20,-1,-5};
int f(int n)
{if(n==0) return 0;
 else
 return
 v[n-1]<v[n] || f(n-1);}

```

Limbaajul Pascal

```

var v:array[0..5] of
integer=(15,12,7,20,-1,-5);
function
f(n:integer):boolean;
begin
  if n=0 then f:=false

```

Limbaajul C

```
int v[]={15,12,7,20,-1,-5};
int f(int n)
{ if(n==0) return 0;
  else
  return
  v[n-1]<v[n] || f(n-1); }
```

a) -2 b) -3 c) 16 d) 4 e) 24 f) 30

```
else
f:=(v[n-1]<v[n]) or f(n-1);
end;
```

13. Se consideră graful neorientat $G=(\{1,2,3,4,5,6\},\{(1,2),(1,3),(1,4),(2,3)\})$.

Precizați care este numărul grafurilor parțiale ale grafului G ?

a) 10 b) 12 c) 8 d) 16 e) 2 f) 4

14. Precizați de câte ori se va executa instrucțiunea de afișare (cout, printf sau write) în secvența de cod de mai jos?

Limbaajul C++

```
for(i=1;i<=10;i++)
  for(j=1;j<=i;j++)
    for(k=1;k<=j;k++)
      cout<<i+j+k;
```

Limbaajul Pascal

```
for i:=1 to 10 do
  for j:=1 to i do
    for k:=1 to j do
      write(i+j+k);
```

Limbaajul C

```
for(i=1;i<=10;i++)
  for(j=1;j<=i;j++)
    for(k=1;k<=j;k++)
      printf("%d",i+j+k);
```

a) 220 b) 110 c) 100 d) 55 e) 150 f) 200

15. Dorim să criptăm un cuvânt scris cu litere mari astfel: fiecare literă este codificată prin codul ei la care se adaugă un număr k ($k \geq 0$). Numărul k se numește cheie de criptare. De exemplu, dacă avem litera **C** și k este **6**, vom obține după criptare litera **I**. Vom considera literele așezate pe un cerc, după **Z** vine **A**. Presupunem că șirul inițial este reținut în variabila *sir* iar rezultatul obținut în urma criptării tot în variabila *sir*.

Considerând prima parte a programului cea de mai jos, precizați care dintre următoarele secvențe realizează criptarea corectă a unui șir de caractere citit de la tastatură cu o cheie k citită de la tastatură?

Limbaajul C++

```
char sir[255];
unsigned int k,i;
cin>>sir;
cin>>k;
```

Limbaajul C

```
char sir[255];
unsigned int k,i;
scanf("%s",sir);
scanf("%u",&k);
```

Limbaajul Pascal

```
var sir:string;
    i,k:word;
begin
  read(sir);
  read(k);
```

Limbaajul C++/C

a) `for(i=0;i<strlen(sir);i++) sir[i]=sir[i]+k;`
 b) `for(i=0;i<strlen(sir);i++) sir[i]=sir[i+k-'A'];`

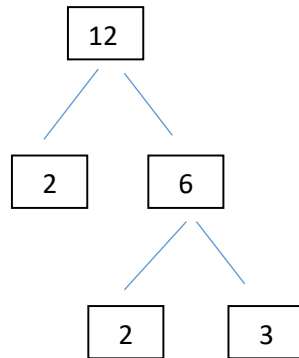
- c) `for(i=0;i<strlen(sir);i++) sir[i]=sir['Z'-'A'+k];`
- d) `for(i=0;i<strlen(sir);i++)`
`sir[i]='A'+(sir[i]-'A'+k)%('Z'-'A'+1);`
- e) `for(i=0;i<strlen(sir);i++)`
`sir[i]='A'+(sir[i]-'A'+'Z'-'A')%('Z'-'A'+1);`
- f) `for(i=0;i<strlen(sir);i++)`
`sir[i]='A'+(sir[i]-'A'+k)%('Z'-'A'+k);`

Limbajul Pascal

- a) `for i:=1 to length(sir) do`
`sir[i]:=chr(ord(sir[i])+k);`
- b) `for i:=1 to length(sir) do`
`sir[i]:=chr(ord(sir[i+k-ord('A')]));`
- c) `for i:=1 to length(sir) do`
`sir[i]:=chr(ord(sir[ord('Z')-ord('A')+k]));`
- d) `for i:=1 to length(sir) do`
`sir[i]:=chr(ord('A')+(ord(sir[i])-ord('A')+k) mod`
`(ord('Z')-ord('A')+1));`
- e) `for i:=1 to length(sir) do`
`sir[i]:=chr(ord('A')+(ord(sir[i])- ord('A')+`
`ord('Z')-ord('A')) mod (ord('Z')- ord('A')+1));`
- f) `for i:=1 to length(sir) do`
`sir[i]:=chr(ord('A')+(ord(sir[i])- ord('A')+k)`
`mod (ord('Z')-ord('A')+k));`

Varianta 22

1. Considerăm descompunerea în factori primi a unui număr reprezentată prin intermediul unui arbore, ca în exemplul de mai jos.



Folosind aceeași reprezentare, precizați câte frunze are arborele obținut pentru numărul **4800**.

- a) 10 b) 12 c) 9 d) 15 e) 6 f) 16

2. Precizați ce se va afișa în urma execuției următorului program?

Limbajul C++

```

#include <iostream>
using namespace std;
int
v[]={5,2,9,11,22,17},n=6;
int calcul(int n)
{int i,poz,max;
max=v[0]; poz=0;
for(i=1;i<n;i++)
if(v[i]>max)
{max=v[i]; poz=i;}
return poz;}
int main()
{ cout<<v[calcul(n)];
return 0; }
```

Limbajul C

```

#include <stdio.h>
int
v[]={5,2,9,11,22,17},n=6;
int calcul(int n)
{ int i,poz,max;
max=v[0];poz=0;
for(i=1;i<n;i++)
if(v[i]>max)
{ max=v[i]; poz=i;}
return poz; }
```

Limbajul Pascal

```

var n:integer;
v:array[0..5] of
integer=(5,2,9,11,22,17);

function
calcul(n:integer):integer;
var i,poz,max:integer;
begin
max:=v[0]; poz:=0;
for i:=1 to n-1 do
if v[i]>max then
begin
max:=v[i];
poz:=i;
end;
calcul:=poz;
end;

begin
n:=6;
write(v[calcul(n)]);
end.
```

```
int main()
{printf("%d",v[calcul(n)]);
 return 0; }
```

- a) Programul nu va afișa nimic, va genera eroare de compilare deoarece apelul de funcții nu este permis ca indice într-un tablou unidimensional. b) 22 c) 17 d) 2 e) 11 f) 9

3. Pentru care dintre următoarele apeluri funcția f va returna valoarea 1?

Limbaajul C/C++

```
int f(int n)
{
    int m=0,n1=n;
    while(n!=0)
    { m=m*10+n%10;
      n=n/10;
    }
    if(m==n1) return 1;
    else return 0;
}
```

Limbaajul Pascal

```
function f(n:integer):integer;
var m,n1:integer;
begin
    m:=0; n1:=n;
    while n<>0 do
    begin
        m:=m*10+n mod 10; n:=n div 10;
    end;
    if m=n1 then f:=1
    else f:=0; end;
```

- a) f(123) b) f(24) c) f(2112) d) f(17) e) f(75) f) f(1592)

4. Într-un plan cartezian se găsesc 8 roboți dați prin coordonatele lor. Coordonatele celor 8 roboți sunt: (2,2), (2,4), (2,6), (2,8), (4,2), (6,2), (6,-2), (-2,-2). Doi roboți A și B se numesc vecini dacă se găsesc pe o paralelă la axele de coordonate și nu există un robot C situat între A și B pe aceeași paralelă. Ținând cont de regulile de mai sus, construți un graf neorientat cu 8 noduri. Nodurile grafului sunt numerotate în ordinea coordonatelor (nodul 1 are coordonatele (2,2), nodul 2 coordonatele (2,4) ș.a.m.d.). Dacă robotul 1 este vecin cu robotul 2 atunci va exista muchie de la 1 la 2 în graf.

Precizați care dintre următoarele afirmații despre graful astfel obținut este adevărată?

- | | | | | | |
|--------------------------|-----------------------------|-------------------------|-------------------------|---|---|
| a) Graful este eulerian. | b) Graful este hamiltonian. | c) Graful este complet. | d) Graful este aciclic. | e) Graful conține două componente conexe. | f) Graful conține trei componente conexe. |
|--------------------------|-----------------------------|-------------------------|-------------------------|---|---|

5. Corectați secvența de program de mai jos astfel încât ea să reprezinte calculul corect al valorii polinomului $P(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ într-un punct c dat.

Limbaajul C/C++

```
p=0;
for(i=0;i<=n;i++)
    p=p*c+a[i];
```

Limbaajul Pascal

```
p:=0;
for i:=0 to n do
    p:=p*c+a[i];
```

Limbaajul C/C++

- a) Instrucțiunea $p=p*c+a[i]$; trebuie

Limbaajul Pascal

înlocuită cu **p=p*a[i]+c;**

b) Instrucțiunea **for (i=0;i<=n;i++)** trebuie înlocuită cu

for (i=n;i>=0;i--)

c) Instrucțiunea **p=p*c+a[i];**

trebuie înlocuită cu **p=c*a[i]+p;**

d) Variabila **p** trebuie inițializată cu -1.

e) Variabila **p** trebuie inițializată cu 1.

f) Instrucțiunea **for (i=0;i<=n;i++)** trebuie înlocuită cu

for (i=1;i<=n;i++)

a) Instrucțiunea **p:=p*c+a[i];** trebuie înlocuită cu **p:=p*a[i]+c;**

b) Instrucțiunea **for i:=0 to n do** trebuie înlocuită cu

for i:=n downto 0 do

c) Instrucțiunea **p:=p*c+a[i];**

trebuie înlocuită cu

p:=c*a[i]+p;

d) Variabila **p** trebuie inițializată cu -1.

e) Variabila **p** trebuie inițializată cu 1.

f) Instrucțiunea **for i:=0 to n do** trebuie înlocuită cu

for i:=1 to n do

6. Precizați ce se va afișa pe ecran în urma execuției următorului program?

Limbajul C++

```
#include <iostream>
using namespace std;
int i;
int g(int i)
{ return i+3; }
int f(int i)
{ return i+g(i); }
int main()
{ i=3;
  cout<<i+f(g(i));
  return 0; }
```

Limbajul Pascal

```
var i:integer;
function g(i:integer):integer;
begin
  g:=i+3;
end;
function f(i:integer):integer;
begin
  f:=i+g(i);
end;
begin
  i:=3; write(i+f(g(i))); end.
```

Limbajul C

```
#include <stdio.h>
int i;
int g(int i)
{ return i+3; }
int f(int i)
{ return i+g(i); }
int main()
{ i=3; printf("%d",i+f(g(i))); return 0; }
```

a) 18

b) 6

c) 15

d) 0

e) 9

f) 12

7. Precizați care dintre următoarele șiruri de numere poate reprezenta șirul gradelor nodurilor unui graf neorientat cu 7 noduri?

a) (1,1,0,1,2,1,6) b) (1,1,1,2,2,5,6) c) (1,1,1,1,2,1,6)

d) (3,3,3,3,3,3,6) e) (1,2,0,0,2,1,5) f) (0,0,0,1,1,2,3)

8. Robotul Robi se află într-un plan cartezian și are în jurul lui un grup de n roboți. El “mănâncă” robotul cel mai apropiat din grupul de roboți. Atât robotul Robi cât și ceilalți

roboți sunt identificați prin coordonatele lor. Datele de intrare se citesc de la tastatură astfel: *n*, numărul roboților din grup, apoi coordonatele robotului Robi și în continuare coordonatele celor *n* roboți. Corecțai programul de mai jos astfel încât să afișeze corect coordonatele robotului care va fi “mâncat”. În cazul în care există mai mulți roboți la aceeași distanță față de Robi, este mâncat primul robot întâlnit la distanța respectivă.

Limbajul C++

```
#include <iostream>
#include <cmath>
using namespace std;
struct robot{
    float x,y;
};
float distanta (robot r1,
               robot r2)
{
    return sqrt((r1.x-r2.x)*
               (r1.x-r2.x)+
               (r1.y-r2.y)*(r1.y-
r2.y));
}
int gaseste(robot robi,
robot a[], int n)
{
    float min,d;
    int i,poz;
    min=distanta(robi,a[0]);
    poz=1;
    for(i=1;i<n;i++)
    {
        d=distanta(robi,a[i]);
        if(d<min){
            min=d;
            poz=i;
        }
    }
    return poz;
}
int main()
{
    int n,i,k;
    robot robi,a[21];
    cin>>n;
    cin>>robi.x>>robi.y;
    for(i=0;i<n;i++)
        cin>>a[i].x>>a[i].y;
    k=gaseste(robi,a,n);
```

Limbajul Pascal

```
type robot=record
    x,y:real;
end;
vector=array[0..20] of
robot;
var a:vector;
    robi:robot;
    n,i,k:integer;

function
distanta(r1,r2:robot):real;
begin
    distanta:=
    sqrt((r1.x-r2.x)*(r1.x-
r2.x)+
        (r1.y-r2.y)*(r1.y-
r2.y));
end;
function gaseste(robi:robot;
a:vector;n:integer):integer;
var min,d:real;
    i,poz:integer;
begin
    min:=distanta(robi,a[0]);
    poz:=1;
    for i:=1 to n-1 do
    begin
        d:=distanta(robi,a[i]);
        if d<min then
            begin
                min:=d;
                poz:=i;
            end;
        end;
    end;
    gaseste:=poz;
end;
begin
    read(n);
    read(robi.x,robi.y);
    for i:=0 to n-1 do
        readln(a[i].x,a[i].y);
```

```

        cout<<a[k].x<<"
"<<a[k].y;
        return 0;
}

```

```

        k:=gaseste(robi,a,n);
        write(a[k].x,' ',a[k].y);
end.

```

Limbajul C

```

#include <stdio.h>
#include <math.h>
typedef struct {
    float x,y;
}robot;

float distanta (robot r1, robot r2)
{
    return sqrt((r1.x-r2.x)*
        (r1.x-r2.x)+
        (r1.y-r2.y)*(r1.y-r2.y));
}

int gaseste(robot robi,
    robot a[],int n)
{
    float min,d;
    int i,poz;
    min=distanta(robi,a[0]);
    poz=1;
    for(i=1;i<n;i++)
    {
        d=distanta(robi,a[i]);
        if(d<min){
            min=d;
            poz=i;
        }
    }
    return poz;
}

int main()
{
    int n,i,k;
    robot robi,a[21];
    scanf("%d",&n);
    scanf("%f %f",&robi.x,&robi.y);
    for(i=0;i<n;i++)
        scanf("%f %f", &a[i].x, &a[i].y);
    k=gaseste(robi,a,n);
    printf("%f %f",a[k].x,a[k].y);
    return 0;}

```

Limbajul C++

- a) Instrucțiunea `if(d<min)` trebuie înlocuită cu `if(d>min)`
- b) Instrucțiunea `min=d;` trebuie înlocuită cu `d=min;`
- c) Instrucțiunea `min=distanța(robi,a[0]);` trebuie înlocuită cu `min=distanța(robi,a[1]);`
- d) Instrucțiunea `poz=1;` trebuie înlocuită cu `poz=0;`
- e) Secvența de instrucțiuni
`for(i=0;i<n;i++) scanf("%f%f",&a[i].x,&a[i].y);`
trebuie înlocuită cu
`for(i=0;i<n;i++) scanf("%f%f",&a.x[i],&a.y[i]);`
- f) Instrucțiunea `poz=i;` trebuie înlocuită cu `i=poz;`

Limbajul C

- a) Instrucțiunea `if(d<min)` trebuie înlocuită cu `if(d>min)`
- b) Instrucțiunea `min=d;` trebuie înlocuită cu `d=min;`
- c) Instrucțiunea `min=distanța(robi,a[0]);` trebuie înlocuită cu `min=distanța(robi,a[1]);`
- d) Instrucțiunea `poz=1;` trebuie înlocuită cu `poz=0;`
- e) Secvența de instrucțiuni
`for(i=0;i<n;i++) cin>>a[i].x>>a[i].y;`
trebuie înlocuită cu
`for(i=0;i<n;i++) cin>>a.x[i]>>a.y[i];`
- f) Instrucțiunea `poz=i;` trebuie înlocuită cu `i=poz;`

Limbajul Pascal

- a) Instrucțiunea `if d<min then` trebuie înlocuită cu `if d>min then`
- b) Instrucțiunea `min:=d;` trebuie înlocuită cu `d:=min;`
- c) Instrucțiunea `min:=distanța(robi,a[0]);` trebuie înlocuită cu `min:=distanța(robi,a[1]);`
- d) Instrucțiunea `poz:=1;` trebuie înlocuită cu `poz:=0;`
- e) Secvența de instrucțiuni
`for i:=0 to n-1 do readln(a[i].x, a[i].y);`
trebuie înlocuită cu
`for i:=0 to n-1 do readln(a.x[i], a.y[i]);`
- f) Instrucțiunea `poz:=i;` trebuie înlocuită cu `i:=poz;`

9. Precizați ce se va afișa pe ecran în urma execuției următorului program?

Limbajul C++

```
#include <iostream>
using namespace std;
int f()
{cout<<1<<" ";
  return 3; }
int main()
{ int s=0,i;
  for(i=1;i<=f();i++)
```

Limbajul C

```
#include <stdio.h>
int f()
{ printf("1 ");
  return 3; }
int main()
{ int s=0,i;
  for(i=1;i<=f();i++)
    s=s+i;
```

Limbajul Pascal

```
var s,i:integer;
function f:integer;
begin
  write('1 ');f:=3;
end;
begin
  s:=0;
  for i:=1 to f do
```

```
s=s+i;
cout<<s; return 0;}
```

Limbajul C++

a) Programul nu va afișa nimic, va genera eroare de compilare deoarece nu este permis apelul unei funcții ca valoare finală a contorului.

- b) 1 1 1 1 6
- c) 1 1 1 6
- d) 1 1 6
- e) 1 10
- f) 1 0

```
printf("%d",s);
return 0;}
```

Limbajul C

a) Programul nu va afișa nimic, va genera eroare de compilare deoarece nu este permis apelul unei funcții ca valoare finală a contorului.

- b) 1 1 1 1 6
- c) 1 1 1 6
- d) 1 1 6
- e) 1 10
- f) 1 0

```
s:=s+i;
writeln(s);end.
```

Limbajul Pascal

a) Programul nu va afișa nimic, va genera eroare de compilare deoarece nu este permis apelul unei funcții ca valoare finală a contorului.

- b) 1 6
- c) 1 1 1 6
- d) 1 1 6
- e) 1 10
- f) 1 0

10. Ce se va afișa pe ecran în urma execuției următorului program dacă de la tastatură se citește cuvântul **caiet** ?

Limbajul C++

```
#include <iostream>
#include <string.h>
using namespace std;
int main()
{char a[255];
int i,j,l,n;
cin.get(a,255);i=0;
n=strlen(a);
while(i<n)
if(strchr("aeiou",a[i]))
{ for(l=1;l<=2;l++)
{ n++;
for(j=n;j>i;j--)
a[j]=a[j-1];
}
a[i+1]='p';
a[i+2]=a[i]; i=i+3;
}
else i++;
cout<<a; return 0; }
```

Limbajul Pascal

```
var a:string[255];
i,j,l,n:integer;
begin
readln(a); i:=1;
n:=length(a);
while i<=n do
if pos(a[i],'aeiou')<>0 then
begin
for l:=1 to 2 do
begin
n:=n+1;
for j:=n downto i+1 do
a[j]:=a[j-1];
end;
a[i+1]:='p';a[i+2]:=a[i];
i:=i+3; end
else i:=i+1;
for i:=1 to n do
write(a[i]);
end.
```

Limbajul C

```
#include <stdio.h>
#include <string.h>
int main()
{
    char a[255];
    int i,j,l,n;
    scanf("%s",a); i=0; n=strlen(a);
    while(i<n)
        if(strchr("aeiou",a[i]))
        {
            for(l=1;l<=2;l++)
            {
                n++;
                for(j=n;j>i;j--)
                    a[j]=a[j-1];
            }
            a[i+1]='p'; a[i+2]=a[i]; i=i+3;
        }
        else i++;
    printf("%s",a); return 0;
}
```

- a) cpapipet b) ct c) cpcapaipiepetpt
d) capaipiepet e) tepeipiapac f) pcpaieptp

11. Ce va afișa următoarea funcție la apelul $f(30)$?

Limbajul C/C++

```
int f(int n)
{
    if(n>50) return n-5;
    else
        return f(f(n+7));
}
```

Limbajul Pascal

```
function f(n:integer):integer;
begin
    if n>50 then
        f:=n-5
    else
        f:=f(f(n+7));
end;
```

- a) 47 b) 50 c) 150 d) 42 e) 44 f) 60

12. Fie $f:A \rightarrow B$. Funcția f este injectivă dacă $\forall x,y \in A, x \neq y \Rightarrow f(x) \neq f(y)$

Pentru funcția $f:\{1,2,3\} \rightarrow \{1,2,3,4,5\}$, folosind metoda backtracking, primele trei funcții injective generate sunt:

x: 1 2 3

f(x): 1 2 3

x: 1 2 3

f(x): 1 2 4

x: 1 2 3

f(x): 1 2 5

Precizați care este a zecea funcție generată.

- a) 5 4 3 b) 4 3 2 c) 2 1 3 d) 1 5 2 e) 1 4 5 f) 1 5 3

13. Numărul ciclurilor hamiltoniene distincte ale unui graf neorientat complet **G** cu $n \geq 3$ noduri este egal cu

- a) n b) $\frac{(n-1)!}{2}$ c) $n!$ d) $\frac{n*(n-1)}{2}$ e) $n*(n-1)$ f) $\frac{n}{2}$

14. Precizați ce se va afișa pe ecran în urma execuției următorului program?

Limbajul C++

```
#include <iostream>
using namespace std;
int v[]={15,12,7,20,-1,-5};
int f(int n)
{ cout<<n<<" ";
  if(n==0) return 0;
  else
  return
    v[n-1]<v[n] || f(n-1);
}
int main()
{ cout<<f(5); return 0; }
```

Limbajul Pascal

```
var v:array[0..5] of
integer=(15,12,7,20,-1,-5);

function
f(n:integer):boolean;
begin
  write(n,' ');
  if n=0 then f:=false
  else f:=(v[n-1]<v[n]) or
f(n-1);
end;
begin
  write(f(5));
end.
```

Limbajul C

```
#include <stdio.h>
int v[]={15,12,7,20,-1,-5};
int f(int n)
{
  printf("%d ",n);
  if(n==0) return 0;
  else return v[n-1]<v[n] || f(n-1);
}
int main()
{ printf("%d",f(5)); return 0; }
```

Limbajul C

- a) 5 4 3 2 1 0
b) 5 4 3 2 1 0 1
c) 5 4 3
d) 5 4 3 1
e) 4 3 1
f) 5 4 3 0 0

Limbajul C++

- a) 5 4 3 2 1 0
b) 5 4 3 2 1 0 1
c) 5 4 3
d) 5 4 3 1
e) 4 3 1
f) 5 4 3 0 0

Limbajul Pascal

- a) 5 4 3 2 1 false
b) 5 4 3 2 1 0 true
c) 5 4 3
d) 5 4 3 true
e) 4 3 true
f) 5 4 3 0 false

15. Dorim să decriptăm un cuvânt scris cu litere mari astfel: fiecare literă este codificată prin codul ei din care se scade un număr k ($k \geq 0$). Numărul k se numește cheie de decriptare. De exemplu, dacă avem litera **I** și k este **6**, vom obține după decriptare litera **C**. Vom considera literele așezate pe un cerc, după **Z** vine **A**. Presupunem că șirul inițial este

reținut în variabila sir iar rezultatul obținut în urma decriptării tot în variabila sir. Considerând prima parte a programului cea de mai jos, care dintre următoarele secvențe realizează decriptarea corectă a unui șir de caractere citit de la tastatură cu o cheie k citită de la tastatură?

Limbajul C++

```
char sir[255];
int d;
unsigned int k,i;
cin>>sir;
cin>>k;
```

Limbajul C

```
char sir[255];
int d;
unsigned int k,i;
scanf("%s",sir);
scanf("%u",&k);
```

Limbajul Pascal

```
var sir:string;
    d:integer;
    i,k:word;
begin
    read(sir);
    read(k);
```

Limbajul C++/C

- `d = 'Z'-'A';`
`for(i=0;i<strlen(sir);i++)`
`sir[i]='A'+(sir[i]-'A'+d)%('Z'-'A'+1);`
- `d = k;`
`for(i=0;i<strlen(sir);i++)`
`sir[i]='A'+(sir[i]-'A'+d)%('Z'-'A'+1);`
- `d = 'Z'-'A'+1-k;`
`for(i=0;i<strlen(sir);i++)`
`sir[i]='A'+(sir[i]-'A'-d)%('Z'-'A'+1);`
- `d = 'Z'-'A'+1-k;`
`for(i=0;i<strlen(sir);i++)`
`sir[i]='A'+(sir[i]-'A'+d)%('Z'-'A'+1);`
- `d = k;`
`for(i=0;i<strlen(sir);i++)`
`sir[i]='A'+(sir[i]-'A'+d)%('Z'-'A'+26);`
- `d = 'Z'-'A';`
`for(i=0;i<strlen(sir);i++)`
`sir[i]='A'+(sir[i]-'A'+d)%('Z'-'A'-25);`

Limbajul Pascal

- `d:=ord('Z')-ord('A');`
`for i:=1 to length(sir) do`
`sir[i]:=chr(ord('A')+(ord(sir[i])-ord('A')+d) mod`
`(ord('Z')-ord('A')+1));`
- `d:=k;`
`for i:=1 to length(sir) do`
`sir[i]:=chr(ord('A')+(ord(sir[i])-ord('A')-d) mod`
`(ord('Z')-ord('A')+1));`
- `d:=ord('Z')-ord('A')+1-k;`
`for i:=1 to length(sir) do`
`sir[i]:=chr(ord('A')+(ord(sir[i])-ord('A')-d) mod`
`(ord('Z')-ord('A')+1));`
- `d:=ord('Z')-ord('A')+1-k;`
`for i:=1 to length(sir) do`


```

    sir[i]:=chr(ord('A')+(ord(sir[i])-ord('A')+d) mod
(ord('Z')-ord('A')+1));
e) d:=k;
   for i:=1 to length(sir) do
     sir[i]:=chr(ord('A')+(ord(sir[i])-ord('A')-d) mod
(ord('Z')-ord('A')+26));
f) d:=ord('Z')-ord('A');
   for i:=1 to length(sir) do
     sir[i]:=chr(ord('A')+(ord(sir[i])-ord('A')+d) mod
(ord('Z')-ord('A')-25));

```

Varianta 23

1. Trei variabile de tip întreg au valorile **a=13**, **b=5**, **c=3**. Dintre expresiile următoare, cea care are valoarea **1** (C++/C) respectiv **true** (Pascal) este:

Limbajul C++/C

- a) **a/c*2<5+c*4%5**
- b) **c%b==a%c**
- c) **b+a/10!=b%c*a/c**
- d) **(b>c) && !(b*c%7==2*a-b*b)**
- e) **c*b*10<a*2**
- f) **c/b*b/c==1**

Limbajul Pascal

- a) **a div c *2 < (5+c*4 mod 5)**
- b) **c mod b =a mod c**
- c) **b+a div 10 <>b mod c *a div c**
- d) **(b>c) and not((b*c mod 7)=(2*a-b*b))**
- e) **c mod b*10 < a*2**
- f) **c div b * b div c = 1**

2. Într-un graf neorientat cu **13** noduri, fiecare nod are gradul **d**. Valoarea lui **d** **nu** poate fi:

- a) **2**
- b) **4**
- c) **6**
- d) **8**
- e) **10**
- f) **11**

3. Variabila **i** este de tip întreg. Numărul total al atribuirilor care se execută în urma rulării secvenței următoare este:

Limbajul C++/C

```
i=1;
while(i*i<2020)
    i=i*2;
```

- a) **5**
- b) **6**
- c) **7**

Limbajul Pascal

```
i:=1;
while i*i<2020 do
    i:=i*2;
```

- d) **9**
- e) **11**
- f) **12**

4. Considerând că variabila **s** poate reține un șir cu cel mult **100** de caractere și variabila **i** este de tip întreg, în urma executării următoarei secvențe de instrucțiuni, lungimea efectivă a șirului **s** este:

Limbajul C++/C

```
strcpy(s, "2020+2020=4040");
for(i=0; i<strlen(s); i++)
if(strchr("0123456789", s[i]))
    strcpy(s+i, s+i+1);
```

- a) **0**
- b) **2**
- c) **5**

Limbajul Pascal

```
s:='2020+2020=4040';
for i:=1 to length(s) do
    if pos(s[i], '0123456789')<>0
then delete(s,i,1);
```

- d) **6**
- e) **8**
- f) **11**

5. Pentru a verifica dacă elementele unui tablou unidimensional cu **n** elemente numere întregi sunt distincte două câte două, numărul de comparații executate este:

- a) **2n**
- b) **n(n-1)/2**
- c) **n(n-1)**
- d) **(n-1)²**
- e) **n²**
- f) **n!**

6. În urma executării următoarei secvențe de program, variabila **x**, de tip întreg, va avea valoarea:

Limbajul C++/C

```
x=15;
x=x*3/4*4/3;
do {if(x%2==0) x=x/2;
    else x=x-5;
}while(x>0);
```

Limbajul Pascal

```
x:=15;
x:=x*3 div 4*4 div 3;
repeat
if x mod 2=0 then x:=x div 2
else x:=x-5;
until x<=0;
```

- a) -6 b) -5 c) -4 d) 0 e) 2 f) 5

7. Utilizând un algoritm backtracking se generează în ordine crescătoare toate numerele naturale cu patru cifre care au suma cifrelor egală cu **4**. Primele trei soluții sunt: **1003**, **1012**, **1021**. În șirul generat, numărul **2020** ocupă poziția:

- a) 10 b) 11 c) 12 d) 13 e) 14 f) 15

8. Dacă **s**, **i**, **j**, **n** sunt variabile de tip întreg și **a** este un tablou bidimensional cu **n** linii și **n** coloane numerotate de la **1** la **n**, următorul algoritm calculează:

Limbajul C++/C

```
s=0;
for (i=1; i<=n; i++)
    for (j=1; j<i; j++)
        s=s+a[i][j];
```

Limbajul Pascal

```
s:=0;
for i:=1 to n do
    for j:=1 to i-1 do
        s:=s+a[i,j];
```

- a) suma elementelor de sub diagonala principală exclusiv elementele diagonalei principale
b) suma elementelor de sub diagonala secundară exclusiv elementele diagonalei secundare
c) numărul elementelor de deasupra diagonalei principale inclusiv elementele diagonalei principale
d) suma elementelor de pe diagonala principală
e) suma elementelor de sub diagonala principală inclusiv elementele diagonalei principale
f) suma elementelor de deasupra diagonalei secundare inclusiv elementele diagonalei secundare

9. Se consideră următoarele declarații de tipuri și variabile:

Limbajul C++

```
struct a
{ int b;
  char c[10];
};
struct d
{ char e[10];
  float f;
  a g;
```

Limbajul Pascal

```
type a=record
  b:integer;
  c:string[10]
end;
d=record
  e: string[10];
  f: real;
  g:a
```

<pre> } h; Limbajul C typedef struct { int b; char c[10]; }a; typedef struct { char e[10]; float f; a g; }d; d h; </pre>	<pre> end; var h:d; </pre>
---	----------------------------

Dintre următoarele expresii, de tip caracter este:

- | | | |
|------------------|--------------------|--------------------|
| a) g.e[2] | b) h.a.c | c) h.a.c[0] |
| d) h.c[2] | e) h.g.c[2] | f) d.e[2] |

10. Se consideră un arbore cu 8 noduri și muchiile [1,2], [2,3], [3,6], [4,3], [5,7], [7,2], [8,2]. Pentru ca arborele să conțină un număr maxim de lanțuri elementare de lungime 3 care nu conțin rădăcina, se poate alege ca rădăcină **oricare** dintre nodurile:

- | | | |
|------------------|------------------|------------------|
| a) 1, 2, 4, 5 | b) 1, 2, 5, 6 | c) 1, 3, 6, 7 |
| d) 2, 3, 4, 5, 6 | e) 1, 4, 5, 6, 8 | f) 2, 3, 4, 7, 8 |

11. În urma executării următoarei secvențe de program tabloul unidimensional **a**, cu 6 elemente numerotate de la 1 la 6, va conține valorile:

<pre> Limbajul C++/C for (i=1;i<=6;i++) if (i%2!=0) a[i]=i/2; else a[i]=7-i; for(i=6;i>=3;i--) a[a[i]]=2*i%7; </pre>	<pre> Limbajul Pascal for i:=1 to 6 do if i mod 2<>0 then a[i]:=i div 2 else a[i]:=7-i; for i:=6 downto 3 do a[a[i]]:=2*i mod 7; </pre>	
a) 0 5 6 1 3 5	b) 5 3 1 3 2 1	c) 6 3 6 2 2 3
d) 6 1 5 2 6 3	e) 6 3 1 2 2 1	f) 6 3 1 3 2 1

12. Pentru apelul **s(2020,2)** al subprogramului de mai jos, enunțul adevărat este:

<pre> Limbajul C++/C int s(int n, int d) { if(n==1) return 0; if (n%d==0) return 1+s(n/d,d); else return s(n,d+1); } </pre>	<pre> Limbajul Pascal function s(n,d:integer): integer; begin if n=1 then s:=0 else if n mod d=0 then s:=1+s(n div d,d) else s:=s(n,d+1) end; </pre>
---	--

- a) **s(2020,2)=3** și reprezintă numărul divizorilor primi ai numărului 2020

- b) $s(2020, 2) = 4$ și reprezintă numărul divizorilor primi ai numărului 2020
 c) $s(2020, 2) = 4$ și reprezintă suma exponenților divizorilor primi din descompunerea în factori primi a numărului 2020
 d) $s(2020, 2) = 6$ și reprezintă suma exponenților divizorilor primi din descompunerea în factori primi a numărului 2020
 e) $s(2020, 2) = 10$ și reprezintă numărul divizorilor proprii ai numărului 2020
 f) $s(2020, 2) = 12$ și reprezintă numărul divizorilor numărului 2020

13. Numărul de grafuri neorientate cu șase noduri, în care nodul 1 are gradul 1 și nodul 2 are gradul 2 este:

- a) 92 b) 1280 c) 1536 d) 1792 e) 1920 f) 2560

14. În urma rulării programului următor vor fi afișate valorile:

Limbajul C++

```
#include <iostream>
using namespace std;
void f (int &a, int b)
{ int x=3;
  a--;
  b++;
  x--;
  cout<<a<<' '<<b<<' '<<x<<' ';
}
int main()
{ int i, x=4, y=6;
  for(i=1; i<=3; i++)
    f(x, x+y);
  cout<<x<<' '<<y;
  return 0;
}
```

Limbajul C

```
#include <stdio.h>
#include <stdlib.h>
void f (int *a, int b)
{ int x=3;
  (*a)--;
  b++;
  x--;
  printf("%d %d %d ", *a, b, x);
}
int main()
{ int i, x=4, y=6;
  for(i=1; i<=3; i++)
    f(&x, x+y);
  printf("%d %d", x, y);
  return 0;
}
```

Limbajul Pascal

```
program main;
var x, y, i: integer;
procedure f (var a: integer; b: integer);
var x: integer;
begin
  x:=3;
  dec(a); inc(b); dec(x);
  write(a, ' ', b, ' ', x, ' ');
end;
begin
  x:=4;
  y:=6;
  for i:=1 to 3 do
    f(x, x+y);
  write(x, ' ', y)
end.
```

a) 3 11 2 3 6

b) 3 11 2 4 6

- c) 3 11 3 2 10 2 1 9 1 1 6 d) 2 11 2 0 9 0 -2 7 -2 -2 6
 e) 3 11 2 3 11 2 3 11 2 4 6 f) 3 11 2 2 10 2 1 9 2 1 6

15. Se sortează crescător tabloul $v = (3, 4, 2, 5, 1, 7, 6)$. O propoziție **falsă** este:
- a) Sortând prin metoda Bubble Sort se fac **7** interschimbări.
 - b) Aplicând metoda de sortare prin interclasare numerele **1** și **4** nu se compară.
 - c) Aplicând metoda de sortare prin selecție se execută cel mult **6** interschimbări.
 - d) Sortând prin selecția minimului, numerele **2** și **3** se compară de două ori.
 - e) Aplicând metoda de sortare Bubble Sort se poate obține ca etapă intermediară tabloul $v = (3, 2, 4, 1, 5, 6, 7)$.
 - f) Aplicând metoda de sortare prin inserție se poate obține ca etapă intermediară tabloul $v = (1, 3, 4, 2, 5, 7, 6)$.

Varianta 24

1. Expresia de mai jos are valoarea **1** (C++/C) respectiv **true** (Pascal) dacă și numai dacă **n** este:

Limbaajul C++/C

`n%2==1 && n*n<100`

Limbaajul Pascal

`(n mod 2=1) and (n*n<100)`

- a) număr întreg impar mai mic decât **10**
- b) număr întreg impar, din intervalul **(-10,10)**
- c) număr natural mai mic decât **100**
- d) număr natural impar de o singură cifră
- e) număr întreg par mai mic decât **10**
- f) număr natural impar cu cel mult două cifre

2. Dacă **a** este un tablou bidimensional cu **n** linii și **n** coloane, numerotate de la **1** la **n**, elementul de pe linia **i** și coloana **j** se află pe diagonala secundară dacă între indici există relația:

a) **`i<j`**

b) **`i>j`**

c) **`i=j`**

d) **`i+j=n-1`**

e) **`i+j=n`**

f) **`i+j=n+1`**

3. Graful neorientat complet **G** are **10** noduri. Un enunț adevărat este:

- a) **G** este arbore
- b) **G** are 50 de muchii
- c) **G** nu este graf hamiltonian și nici eulerian
- d) **G** este graf hamiltonian dar nu eulerian
- e) **G** nu este graf hamiltonian dar este graf eulerian
- f) **G** este graf hamiltonian și eulerian

4. Se consideră că **d, i, k, n** sunt variabile de tip întreg și **a** este un tablou unidimensional cu **n** numere întregi numerotate de la **1** la **n**. La finalul execuției secvenței următoare, variabila **k** are valoarea **1** dacă și numai dacă elementele tabloului **a** formează o progresie aritmetică. Expresia corectă care completează punctele de suspensie este:

Limbaajul C++/C

`k=1;`

`d=a[2]-a[1];`

`for (i=3; i<=n; i++)`

`if ( . . . )`

`k=0;`

Limbaajul Pascal

`k:=1;`

`d:=a[2]-a[1];`

`for i:=3 to n do`

`if . . . then`

`k:=0;`

Limbaajul C++/C

a) **`a[i+1]-a[i]!=d`**

b) **`a[i]-a[i+1]!=d`**

c) **`a[i]-a[i-1]!=d`**

d) **`a[i+1]-a[i]==d`**

e) **`a[i]+a[i+1]!=d`**

f) **`a[i]-a[i-1]==d`**

Limbaajul Pascal

a) **`a[i+1]-a[i]<>d`**

b) **`a[i]-a[i+1] <>d`**

c) **`a[i]-a[i-1]<>d`**

d) **`a[i+1]-a[i]=d`**

e) **`a[i]+a[i+1]<>d`**

f) **`a[i]-a[i-1]=d`**

5. Se consideră un arbore cu 8 noduri și muchiile [1,2], [2,3], [3,6], [4,3], [5,7], [7,2], [8,2]. Înălțimea arborelui este egală cu lungimea celui mai lung lanț elementar care unește rădăcina de o frunză. Arborele dat are înălțime minimă dacă se va alege ca rădăcină nodul:

a) 1 b) 2 c) 3 d) 5 e) 7 f) 8

6. În urma execuției secvenței următoare, în care toate variabilele sunt de tip întreg, valoarea variabilei **n** este:

Limbajul C++/C

```
n=0; a=11357; b=1426; p=1;
while(a!=b)
{ x=a%10;y=b%10;
  if(x<y) n=n+p*x;
  else n=n+p*y;
  p=p*10;a=a/10;b=b/10;
}
```

Limbajul Pascal

```
n:=0; a:=11357; b:=1426; p:=1;
while a<>b do begin
x:=a mod 10; y:=b mod 10;
if x<y then n:=n+p*x
else n:=n+p*y;
p:=p*10; a:=a div 10; b:=b
div 10
end;
```

a) 1326 b) 1356 c) 6241
d) 11326 e) 11457 f) 62411

7. Fie enunțul: „pentru a sorta descrescător un tablou unidimensional cu 20 de elemente numere reale, utilizând metoda selecției, nu sunt necesare mai mult de **x** determinări ale valorii maxime”. Enunțul este adevărat dacă **x** este egal cu:

a) 0 b) 10 c) 19 d) 20 e) 190 f) 400

8. Matricea alăturată este matricea de adiacență a unui graf:

```
0 1 1 0
1 0 1 0
1 0 0 0
0 1 0 0
```

a) orientat cu 6 noduri și 3 arce b) neorientat cu 4 noduri și 3 muchii
c) orientat cu 4 noduri și 6 arce d) neorientat cu 6 noduri și 6 muchii
e) orientat cu 4 noduri și 3 arce f) neorientat cu 4 noduri și 6 muchii

9. Utilizând un algoritm backtracking se generează în ordine lexicografică toate anagramele cuvântului **roman**. Soluția generată imediat înainte de cuvântul **norma** și soluția generată imediat după cuvântul **norma** sunt:

a) **nramo** și **noram** b) **nramo** și **nrmao** c) **nomra** și **noram**
d) **nomra** și **nramo** e) **noram** și **nramo** f) **nomar** și **nramo**

10. Variabilele **i**, **j**, **k** sunt de tip întreg iar **s** reține un șir de caractere format din litere mici și spații (cuvintele sunt despărțite printr-un singur spațiu). În urma executării următoarei secvențe de program, variabila **k** are valoarea 0 dacă șirul **s** este inițial:

Limbajul C++/C

```
for(i=0;i<strlen(s);i++)
if(s[i]==' ')
strcpy(s+i,s+i+1);
```

Limbajul Pascal

```
for i:=1 to length(s) do
if s[i]=' ' then
delete(s,i,1);
```



```

i:=0;
j:=strlen(s)-1;
k:=1;
while(i<j)
{
    if (s[i]!=s[j])
        k:=0;
    i++;
    j--;
}

```

a) atasata
d) elisa vasile

```

i:=1;
j:=length(s);
k:=1;
while i<j do
begin
    if s[i]<>s[j] then
        k:=0;
    inc(i);
    dec(j)
end;

```

b) o rama maro
e) nora aron

c) o rama alba
f) vasile elisav

11. Dacă din programul principal se apelează $f(f(3))$, numărul de autoapeluri ale funcției f , definită mai jos, este:

Limbajul C++/C

```

int f (int a)
{
    if (a<2)
        return 1;
    else
        return f(a-1)+2*f(a-3);
}

```

a) 8

b) 9

c) 10

Limbajul Pascal

```

function f(a:integer):integer;
begin
    if a<2 then
        f:=1
    else
        f:=f(a-1)+2*f(a-3)
end;

```

d) 14

e) 15

f) 16

12. Secvența de mai jos construiește tabloul bidimensional a cu n linii și n coloane, numerotate de la 1 la n . Pentru $n=4$, suma elementelor de pe diagonala principală este:

Limbajul C++/C

```

x:=1;
y:=1;
for(i:=1;i<=n;i++)
    for(j:=1;j<=n+1-i;j++)
    {
        a[i][j]=x;
        x++;
    }
for(j:=n;j>=1;j--)
    for(i:=n;i>=n+1-j;i--)
    {
        a[i][j]=y;
        y++;
    }

```

a) 9

b) 12

c) 14

Limbajul Pascal

```

x:=1;
y:=1;
for i:=1 to n do
    for j:=1 to n+1-i do
        begin
            a[i,j]:=x;
            inc(x)
        end;
for j:=n downto 1 do
    for i:=n downto n+1-j do
        begin
            a[i,j]:=y;
            inc(y)
        end;

```

d) 16

e) 28

f) 30

13. Pentru funcția dată mai jos, $f(95)$ și $f(59)$ au valorile:

Limbajul C++/C

```

int f (int x)
{
    if (x>=100)

```

Limbajul Pascal

```

function f (x:integer) :
integer;
begin

```

```

        return x+2;
    else
        return f (f(x+2)+1);
}

```

```

    if x>=100 then
        f:=x+2
    else
        f:=f(f(x+2)+1)
    end;

```

- | | | |
|---------------|---------------|---------------|
| a) 103 și 146 | b) 109 și 162 | c) 110 și 163 |
| d) 103 și 163 | e) 112 și 157 | f) 112 și 166 |

14. Sortând crescător prin metoda selecției, cu număr minim de interschimbări (se interschimbă doar elemente distincte), tablourile unidimensionale $\mathbf{v}=(3, 8, 2, 7)$, $\mathbf{x}=(4, 5, 1, 7)$, $\mathbf{y}=(4, 7, 9, 6)$ și $\mathbf{z}=(6, 3, 2, 9)$ se calculează numărul operațiilor (comparări și atribuirii) efectuate. Afirmatia adevărată este:

- a) Pentru $\mathbf{v}$ și $\mathbf{y}$ s-a realizat un număr egal de operații
- b) Pentru $\mathbf{v}$ și $\mathbf{z}$ s-a realizat un număr egal de operații
- c) Cel mai mare număr de operații s-a efectuat pentru $\mathbf{x}$
- d) Cel mai mare număr de operații s-a efectuat pentru $\mathbf{y}$
- e) Cel mai mic număr de operații s-a efectuat pentru $\mathbf{z}$
- f) Cel mai mic număr de operații s-a efectuat pentru $\mathbf{y}$

15. În urma executării secvenței de program de mai jos se afișează:

Limbajul C++/C

```

int f (int a, int b, int e)
{
    int x;
    if(a<2)
        return e+1;
    if(a%b==0)
    {
        if(e==0)
            cout<<b<<' ';
        |printf("%d ",b);
        e++;
        return f(a/b,b,e);
    }
    else
    {
        x=e+1;
        e=0;
        b++;
        return x*f(a,b,e);
    }
}
int main()
{ int x,y,e;
  cin>>x;  |scanf("%d",&x);
  y=2;
  e=0;
  cout<<f(x,y,e);
  |printf("%d",f(x,y,e));
}

```

Limbajul Pascal

```

program p;
var x,y,e: integer;
function f(a,b,e:integer) :integer;
var x:integer;
begin
    if a<2 then
        f:=e+1
    else
        begin
            if a mod b=0 then
                begin
                    if e=0 then
                        write(b,' ');
                    inc(e);
                    f:=f(a div b,b,e)
                end
            else
                begin
                    x:=e+1;e:=0;inc(b);
                    f:=x*f(a,b,e)
                end
            end
        end
    end;
end;
begin
    read(x);
    y:=2;
    e:=0;
    writeln(f(x,y,e))
end.

```

} | **end.**

- a) divizorii proprii ai numărului x
- b) numărul de divizori proprii ai numărului x
- c) divizorii proprii și numărul divizorilor proprii ai numărului x
- d) divizorii primi ai lui x și numărul tuturor divizorilor lui x
- e) divizorii proprii ai numărului x și produsul exponenților divizorilor primi din descompunerea în factori primi a numărului x
- f) divizorii primi ai numărului x și produsul exponenților divizorilor primi din descompunerea în factori primi a numărului x

Varianta 25

1. Se dă o variabilă **a** care reține un număr natural nenul. Care dintre următoarele expresii are valoarea **0/false**?
 - a) C++/C: **a*(a+1)/2<a*a+1**
Pascal: **a*(a+1) DIV 2<a*a+1**
 - b) C++/C: **4*a*(a-1)<a*a-2**
Pascal: **4*a*(a-1)<a*a-2**
 - c) C++/C: **a>0 && (a%10+(a+1)%10)**
Pascal: **a>0 AND (a MOD 10+(a+1) MOD 10)**
 - d) C++/C: **a%2+(a+1)%2==1**
Pascal: **a MOD 2+(a+1) MOD 2 = 1**
 - e) C++/C: **(a-1)*(a+1)>a*a-2**
Pascal: **(a-1)*(a+1)>a*a-2**
 - f) C++/C: **a*(a+1)>=a*a+1**
Pascal: **a*(a+1)>=a*a+1**

2. În secvența de program variabilele **i** și **j** sunt de tip întreg iar **A** este un tablou bidimensional cu 5 linii și 5 coloane numerotate de al 0 la 4,

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 0 \\ 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 0 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}. \text{ În urma executării următoarelor instrucțiuni se va afișa:}$$

Limbajul C++

```
i=4;
j=0;
cout<<A[i][j];
while(j!=4)
{
    i--;
    cout<<A[i][j];
    j++;
    cout<<A[i][j];
}
```

Limbajul C

```
i=4;
j=0;
printf("%d",A[i][j]);
while(j!=4)
{
    i--;
    printf("%d",A[i][j]);
    j++;
    printf("%d",A[i][j]);
}
```

Limbajul Pascal

```
i:=4;
j:=0;
write(A[i][j]);
while j<>4 do
begin
    i:=i-1; write(A[i][j]);
    j:=j+1;
```

- write(A[i][j])**
end;
- a) 167238945 b) 127834905 c) 549832761
d) 509438721 e) 127850943 f) 509412783
3. Fie o coadă inițial vidă. Cu ajutorul subprogramelor **Ad(x)** respectiv **El()** este adăugat elementul **x** respectiv șters un element din coadă. Care va fi suma elementelor din coadă în urma executării operațiilor următoare?
Ad(3) El() Ad(7) Ad(9) El() Ad(5) Ad(2) El()
a) 12 b) 14 c) 10 d) 15 e) 7 f) 16
4. Fie $M = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$ matricea de adiacență a unui graf neorientat **G**.
Numărul de componente conexe ale grafului **G** este:
a) 0 b) 1 c) 2 d) 3 e) 4 f) 5
5. Numărul de noduri și numărul de frunze ale arborelui cu rădăcină memorat în următorul vector de tați (**0, 1, 1, 2, 2, 3, 6, 7, 7**) este:
a) 9 0 b) 9 5 c) 8 4 d) 1 2 e) 8 3 f) 9 4
6. Se dă mulțimea **A={1, 2, 3, 4}**. Un algoritm generează în ordine crescătoare, toate numerele naturale de **n** cifre, folosind cifre din mulțimea **A**, numere care au suma cifrelor egală cu **6**. Dacă pentru **n=3**, primele trei soluții generate sunt, în ordine, **114, 123, 132**, numărul de ordine al soluției **312** este:
a) 4 b) 5 c) 6 d) 7 e) 8 f) 9
7. Variabila **i** memorează un număr natural, iar variabila **a** memorează șirul **examen**. În urma executării următoarelor instrucțiuni se va afișa:
- | | |
|--|---|
| <p>Limbajul C++/C</p> <pre>for(i:=0;i<5;i++) if(a[i]<a[i+1]) a[i]=a[i+1]; cout<<a; printf("%s",a);</pre> | <p>Limbajul Pascal</p> <pre>for i:=0 to 4 do if a[i]<a[i+1] then a[i]:=a[i+1]; write(a);</pre> |
|--|---|
- a) **xxmmnn** b) **exmmnn** c) **fxbmfn**
d) **exxxxn** e) **xamenn** f) **nemaxe**
8. Fie subprogramul recursiv:
- | | |
|--|--|
| <p>Limbajul C++/C</p> <pre>void numar(int n) { if(n<=100)</pre> | <p>Limbajul Pascal</p> <pre>procedure numar(n: longint); begin if n<=100 then</pre> |
|--|--|

```

    cout<<'\\n';
    |printf("\\n");
else
{
    if(n%10<5)
        cout<<n%10;
        |printf("%d", n%10);
    numar(n/10);
    if(n%10>5)
        cout<<n%10;
        |printf("%d", n%10);
    }
}

```

```

    writeln
else
begin
    if n mod 10 < 5 then
        write(n mod 10);
        numar(n div 10);
    if n mod 10 > 5 then
        write(n mod 10)
    end
end;

```

În urma apelului **numar(824972345)** se va afișa:

- | | | | | | |
|---------|---------|---------|---------|---------|---------|
| a) 4324 | b) 4324 | c) 4234 | d) 4234 | e) 3244 | f) 2443 |
| 79 | 97 | 79 | 97 | 97 | 97 |

9. În urma executării următoarelor instrucțiuni se va afișa valoarea:

Limbajul C++/C

```

int a=360,b=0,c=2;
while(a!=1)
{
    if(!(a%c))
    {
        while(!(a%c))
            a/=c;
        b++;
    }
    c++;
}
cout<<b; | printf("%d", b);

```

Limbajul Pascal

```

var a,b,c: integer;
begin
    a:=360; b:=0; c:=2;
    while a<>1 do
    begin
        if a mod c=0 then
        begin
            while a mod c=0 do
                a:=a div c;
            b:=b+1
        end;
        c:=c+1
    end;
    write(b)
end.

```

- | | | | | | |
|------|------|------|------|------|------|
| a) 2 | b) 3 | c) 4 | d) 5 | e) 6 | f) 7 |
|------|------|------|------|------|------|

10. Fie tablou unidimensional **v=(5,8,1,3,6,7,4,9)**, elementele fiind numerotate de la 0 la 7. După executarea următoarelor instrucțiuni, tabloul unidimensional **v** va conține valorile:

Limbajul C++/C

```

i=0;
while(i<=3)
{
    if(v[i]<5)
        v[i]=2*v[i];
    if(v[7-i]>v[i])

```

Limbajul Pascal

```

i:=0;
while i<=3 do
begin
    if v[i]<5 then
        v[i]:=2*v[i];
    if v[7-i]>v[i] then

```

<pre> v[7-i]=v[7-i]-v[i]; i=i+1; } </pre>	<pre> v[7-i]:=v[7-i]-v[i]; i:=i+1 end; </pre>
a) v=(5,8,2,6,6,4,4,3)	b) v=(5,8,2,6,6,7,8,9)
c) v=(5,8,2,6,6,5,4,4)	d) v=(5,8,2,6,6,6,4,4)
e) v=(5,8,8,6,6,4,4,4)	f) v=(5,5,2,2,6,6,4,4)

11. Variabilele **s**, **i**, **c** sunt de tip întreg. Variabila **c** memorează un număr natural par. În urma executării secvenței de instrucțiuni, variabila **s** are valoarea:

Limbajul C/C++ <pre> s=0; for(i=1;i<=c/2;i++) s=s+i; s=2*s; cout<<s; printf("%d", s); </pre>	Limbajul Pascal <pre> s:=0; for i:=1 to (c DIV 2) do s:=s+i; s:=2*s; write(s); </pre>
a) $\frac{c \cdot (c+1)}{2}$	b) $c \cdot (c+1)$
c) $\frac{c \cdot (c+1)}{4}$	d) $\frac{c \cdot (c+2)}{4}$
e) $\frac{c \cdot (c-1)}{2}$	f) $\frac{c \cdot (c-1)}{4}$

12. Variabilele **i** și **c** sunt de tip întreg, iar tablou unidimensional **a** are valorile (1, 2, 3, 4, 5, 6, 7), primul element se află pe poziția 1, al doilea element se află pe poziția 2, ș.a.m.d. În urma executării instrucțiunilor, tabloul **a** va conține valorile:

Limbajul C++/C <pre> for(i=1;i<=7;i++) { int c; c=a[i]; a[i]=a[7-i+1]; a[7-i+1]=c; } </pre>	Limbajul Pascal <pre> for i:=1 to 7 do begin c:=a[i]; a[i]:=a[7-i+1]; a[7-i+1]:=c end; </pre>	
a) 1 2 3 4 5 6 7	b) 7 6 5 4 3 2 1	c) 7 6 5 4 5 6 7
d) 1 2 3 4 3 2 1	e) 4 3 2 1 2 3 4	f) 4 5 6 7 6 5 4

13. În urma executării următorului program se va afișa:

Limbajul C++/C <pre> int main() { int c,i,nr=0; for(i=200; i<=300; i++) { c=i; while(c!=0) { </pre>	Limbajul Pascal <pre> var nr, i, c: integer; begin nr:=0; for i:=200 to 300 do begin c:=i; while c<>0 do begin </pre>
---	--

<pre> if(c%2==1) nr++; c=c/10; } } cout<<nr; printf("%d", nr); return 0;} </pre>	<pre> if c mod 2 = 1 then nr:=nr+1; c:=c div 10 end end; write(nr) end. </pre>
--	--

a) 50 b) 99 c) 100 d) 101 e) 102 f) 103

14. Subprogramul **mat** definit mai jos, cu doi parametri:

- **n**, prin care primește un număr natural nenul ($n \leq 100$),
- **d**, prin care primește elementele unui tablou bidimensional cu **n** linii și **n** coloane, numerotate de la 1 la n, determină:

Limbajul C++/C

```

int mat(int n, int d[][100])
{int e[100][100],i,j,k,p,matrice=0;
  if(n==1) {matrice=d[1][1];
            return matrice;}
  else
  {
    for(i=1;i<=n;i++)
    {
      for(k=2;k<=n;k++)
      for(j=1;j<i;j++)
        e[k-1][j]=d[k][j];
      for(k=2;k<=n;k++)
      for(j=i+1;j<=n;j++)
        e[k-1][j-1]=d[k][j];
      if((i+1)%2==0) p=1;
      else p=-1;
      matrice=matrice+p*d[1][i]*mat(n-1,e);
    }
    return matrice;
  }
}

```

Limbajul Pascal

```

type matrix = array[1..100,1..100] of integer;
function mat(n:integer; d:matrix):integer;
var e: matrix;
    i,j,k,p,matrice: integer;
begin
  matrice:=0;
  if n=1 then
  begin
    matrice:=d[1][1];
    mat:=matrice;
  end;
end;

```



```

end
else
begin
  for i:=1 to n do
    begin
      for k:=2 to n do
        for j:=1 to i-1 do
          e[k-1][j]:=d[k][j];
        for k:=2 to n do
          for j:=i+1 to n do
            e[k-1][j-1]:=d[k][j];
          if (i+j) MOD 2 =0 then
            p:=1
          else p:=-1;
          matrice:=matrice+p*d[1][i]*mat(n-1,e);
          end;
        mat:=matrice end end;

```

- | | |
|------------------------------|---------------------------------------|
| a) pătratul matricei | b) transpusa matricei |
| c) determinantul matricei | d) matricea inversă |
| e) înmulțirea a două matrice | f) înmulțirea matricei cu o constantă |

15. În urma executării subprogramului, pentru parametrii **v**, **n**, **k**, cu valorile de intrare

$$v = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 0 & 1 \\ 2 & 3 & 4 & 5 \end{pmatrix}, n=4 \text{ și } k=2*n, \text{ se va afișa:}$$

Limbajul C++/C

```

void afis(int v[][100],int n,int k)
{int i;
if(k!=1)
{
  if(k%2==0){
    for(i=1;i<=n;i++)
      if(k-i<=n && k-i>0)
        cout<<v[i][k-i]; | printf("%d", v[i][k-i]);
  }
  else {
    for(i=n;i>=1;i--)
      if(k-i<=n && k-i>0)
        cout<<v[i][k-i]; | printf("%d", v[i][k-i]);
  }
  afis(v,n,k-1);
}}

```

Limbajul Pascal

```

type matrice = array of array of integer;

```

```

procedure afis(var v:matrice; n:integer; k:integer);
var i: integer;
begin
  if k<>1 then
    begin
      if (k MOD 2)=0 then
        begin
          for i:=1 to n do
            if(k-i<=n) AND (k-i>0) then
              write(v[i][k-i]);
            end
          else
            for i:=n downto 1 do
              if(k-i<=n) AND (k-i>0)
                write(v[i][k-i])
            afis(v,n,k-1)
          end end;

```

a) 5143073692852140
 c) 0148523692307145
 e) 0167093290672145

b) 5417032963258410
 d) 0412582963703415
 f) 5498563245014398

Varianta 26

1. Se dă o variabilă **a** care reține un număr natural nenul. Expresia care are valoarea **0/false** pentru orice număr natural nenul **a** este:

- a) C++/C: $(a/3+a/7)\%9$
Pascal: $(a \text{ DIV } 3+a \text{ DIV } 7) \text{ MOD } 9$
- b) C++/C: $(a\%10+a\%100/10)/10$
Pascal: $((a \text{ MOD } 10)+(a \text{ MOD } 100) \text{ DIV } 10) \text{ DIV } 10$
- c) C++/C: $((10-a\%10)+(10-a\%100/10))/10$
Pascal: $((10-(a \text{ MOD } 10))+(10-(a \text{ MOD } 100) \text{ DIV } 10)) \text{ DIV } 10$
- d) C++/C: $(a\%5+a\%7)/10$
Pascal: $(a \text{ MOD } 4+a \text{ MOD } 6) \text{ DIV } 10$
- e) C++/C: $(a\%3+a\%7)/9$
Pascal: $(a \text{ MOD } 3+a \text{ MOD } 7) \text{ DIV } 9$
- f) C++/C: $(a\%10+a/10)/9$
Pascal: $(a \text{ MOD } 10+a \text{ DIV } 10) \text{ DIV } 9$

2. Se dă un tablou unidimensional **v=(3,5,8,4,2,6,9,1)** în care primul element se află pe poziția 0 și **i** o variabilă de tip întreg. În urma executării secvenței de instrucțiuni, elementele tabloului unidimensional **v** sunt:

Limbajul C++/C

```
i=0;
while(i<=6)
{
j=i+1;
v[i]=v[i]+v[j];
v[j]=v[i]-v[j];
v[i]=v[i]-v[j];
i=i+2;
}
```

Limbajul Pascal

```
i:=0;
while i<=6 do
begin
j:=i+1;
v[i]:=v[i]+v[j];
v[j]:=v[i]-v[j];
v[i]:=v[i]-v[j];
i:=i+2
end;
```

- a) **v=(5,8,4,2,6,9,1,3)**
- b) **v=(5,3,4,8,6,2,1,9)**
- c) **v=(11,3,20,8,10,2,19,9)**
- d) **v=(5,-7,4,0,6,10,1,6)**
- e) **v=(3,1,9,6,2,4,8,5)**
- f) **v=(9,1,2,6,8,4,3,5)**

3. Știind că variabila **i** este de tip întreg și variabila **a** de tip șir de caractere reține cuvântul **politehnica**, în urma executării instrucțiunilor se va afișa:

Limbajul C++/C

```
for(i=0;i<=7;i++)
if(a[i]<'n')
a[i]='A'-'a'+a[i];
cout<<a; | printf("%s", a);
```

Limbajul Pascal

```
for i:=1 to 8 do
begin
if a[i]<'n' then
a[i]:=upcase(a[i])
```

		end; write(a);
a) poLiTEHnICA	b) POliTehnica	c) POliTehnica
d) POliTehNICA	e) poliTEHNICA	f) poLiTEHnica

4. Fie un tablou bidimensional **A**, cu **4** linii și **4** coloane numerotate de la **0** la **3** care conține elemente de tip întreg și două variabile **i** și **j** de tip întreg. Valorile ce vor fi reținute în tabloul bidimensional **A** după executarea următoarelor instrucțiuni sunt:

Limbajul C++/C

```
i=3;
while(i>=0)
{
    j=3;
    while(j>=0)
    {
        if((i+j)%2==0)
            A[i][j]=i+j;
        else
            if(i>j) A[i][j]=i;
            else A[i][j]=j;
        j--;
    }
    i--;
}
```

Limbajul Pascal

```
i:=3;
while i>=0 do
begin
    j:=3;
    while j>=0 do
    begin
        if (i+j) MOD 2 =0 then
            A[i,j]:=i+j
        else
            if i>j then
                A[i,j]:=i
            else
                A[i,j]:=j;
        j:=j-1
    end;
    i:=i-1
end;
```

a) $A = \begin{pmatrix} 0 & 0 & 2 & 0 \\ 0 & 2 & 1 & 4 \\ 2 & 1 & 4 & 2 \\ 0 & 4 & 2 & 6 \end{pmatrix}$

b) $A = \begin{pmatrix} 0 & 1 & 0 & 3 \\ 1 & 1 & 3 & 1 \\ 0 & 3 & 2 & 5 \\ 3 & 1 & 5 & 3 \end{pmatrix}$

c) $A = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 1 & 3 & 3 \\ 2 & 3 & 2 & 5 \\ 3 & 3 & 5 & 3 \end{pmatrix}$

d) $A = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 2 & 4 \\ 2 & 2 & 4 & 3 \\ 3 & 4 & 3 & 6 \end{pmatrix}$

e) $A = \begin{pmatrix} 0 & 2 & 1 & 3 \\ 2 & 1 & 3 & 3 \\ 1 & 3 & 2 & 5 \\ 3 & 3 & 5 & 3 \end{pmatrix}$

f) $A = \begin{pmatrix} 0 & 2 & 1 & 3 \\ 2 & 1 & 3 & 5 \\ 1 & 3 & 5 & 3 \\ 3 & 5 & 3 & 3 \end{pmatrix}$

5. Fie o stivă inițial vidă. Cu ajutorul subprogramelor **Ad(x)**, respectiv **El()** este adăugat elementul **x**, respectiv șters un element din stivă. Suma elementelor din stivă după executarea operațiilor următoare este:

Ad(3) Ad(7) Ad(5) El() El() Ad(8)

a) **3** b) **7** c) **10** d) **11** e) **12** f) **13**

6. Se dă mulțimea $A = \{1, 4, 5, 8, 9\}$. Un algoritm generează în ordine crescătoare, toate numerele naturale de n cifre, folosind cifre distincte din mulțimea A , care nu au alăturate cifre de aceeași paritate. Dacă pentru $n=4$, primele patru soluții generate sunt: **1458, 1498, 1854, 1894**, numărul de soluții pe care le va genera algoritmul este:
- a) 12 b) 16 c) 20 d) 24 e) 28 f) 30
7. Șirul care poate reprezenta valorile gradelor nodurilor unui graf neorientat cu 6 noduri este:
- a) **2 5 0 2 1 2** b) **2 2 1 1 1 2** c) **2 2 7 2 2 1**
d) **2 2 0 0 4 2** e) **2 3 1 1 2 2** f) **2 2 2 1 1 2**
8. Șirul de valori care poate fi vectorul de tați al unui arbore cu 8 noduri este:
- a) $T = (0\ 1\ 8\ 3\ 2\ 5\ 5\ 4)$ b) $T = (0\ 3\ 7\ 2\ 4\ 5\ 8\ 3)$
c) $T = (0\ 1\ 1\ 2\ 4\ 5\ 8\ 7)$ d) $T = (0\ 5\ 7\ 3\ 1\ 3\ 1\ 2)$
e) $T = (0\ 1\ 0\ 2\ 4\ 6\ 3\ 3)$ f) $T = (8\ 5\ 7\ 3\ 1\ 3\ 1\ 2)$
9. Știind că i, j, s și a sunt patru variabile de tip întreg, pentru orice valoare naturală nenulă a variabilei a , după executarea instrucțiunilor, valoarea afișată corespunde formulei matematice:

Limbajul C++/C

```
s=0;
for(i=1;i<=a;i++)
{
    j=1;
    while(j<=i)
    {
        s++;
        j++;
    }
    j=i+1;
    do
    {
        s++;
        j++;
    }while(j<=a);
}
cout<<s--; | printf("%d", s--);
```

Limbajul Pascal

```
s:=0;
for i:=1 to a do
begin
    j:=1;
    while j<=i do
    begin
        s:=s+1;
        j:=j+1
    end;
    j:=i+1;
    repeat
        s:=s+1;
        j:=j+1;
    until j>a;
end;
write(s);
```

- a) $a(a+1)$ b) a^2+1 c) a^2
d) a^2-1 e) $a(a-1)$ f) $2a^2-1$

10. Subprogramul **afis** primește ca parametru un tablou bidimensional **v** cu **n** linii și **n** coloane, numerotate de la 1 la **n**, unde

$$v = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 0 & 1 \\ 2 & 3 & 4 & 5 \end{pmatrix}, n=4 \text{ și } k=2 \cdot n.$$

Pentru valorile date, **afis(v,n,k)** va afișa:

Limbajul C++/C

```
void afis(int v[100][100],int n,int k)
{int i;
if(k>1)
{ for(i=n;i>=1;i--)
    if(k-i<=n && k-i>0)
        cout<<v[i][k-i]; | printf("%d", v[i][k-i]);
afis(v,n,k-2);
}
```

Limbajul Pascal

```
type matrice = array [1..100,1..100] of integer;
procedure afis(var v:matrice; n:integer; k:integer);
var i: integer;
begin
if k<>1 then
begin
for i:=n downto 1 do
    if(k-i<=n) AND (k-i>0) then
        write(v[i,k-i]);
afis(v,n,k-2)
end
end;
```

- | | | |
|-------------|-------------|-------------|
| a) 08523075 | b) 57032580 | c) 02587035 |
| d) 53078520 | e) 35087250 | f) 70358520 |

11. Știind că subprogramul **functie** corespunde funcției matematice **f(x)=3·x-1**, pentru orice **x** număr întreg, **abc(t,c)** va calcula:

Limbajul C++/C

```
int functie(int x)
{return 3*x-1;}

int abc(int t, int c)
{ if(c==0) return t;
else
return abc(functie(t),c-1);}
```

Limbajul Pascal

```
function functie(var x:integer):integer;
begin
    functie:=3*x-1
end;
```

```
function abc(t,c:integer):integer;
begin
    if c=0 then
        abc:=t
    else
        abc:=abc(functie(t),c-1)
    end;
```

a) $\underbrace{f(t) \circ \dots \circ f(t)}_{c-1}$

b) $\underbrace{f(t) + \dots + f(t)}_{c-1}$

c) $f(t)^c$

d) $c * f(t)$

e) $(c - 1) * f(t)$

f) $\underbrace{f(t) \circ \dots \circ f(t)}_c$

12. După executarea următoarelor instrucțiuni se va afișa:

Limbajul C++/C

```
char a[20][20];
int i;
strcpy(a[1], "bacalaureat");
strcpy(a[2], "liceu");
strcpy(a[3], "examene");
strcpy(a[4], "politehnica");
for(i=1; i<=4; i++)
    cout<<a[i][2*i];
| printf("%d", a[i][2*i]);
```

Limbajul Pascal

```
var a:array[1..20] of
string;
    i:integer;
begin
a[1]:='bacalaureat';
a[2]:='liceu';
a[3]:='examene';
a[4]:='politehnica';
for i:=1 to 4 do
    write(a[i,2*i+1])
end.
```

a) aenn

b) teen

c) cunc

d) cuei

e) bceh

f) ceen

13. Următoarele instrucțiuni vor afișa:

Limbajul C++

```
int f1(int x, int &y)
{
    x=x+2;
    y=y-1;
    return x+y;
    x=x+1;
}
```

Limbajul C

```
int f1(int x, int *y)
{
    x=x+2;
    *y=*y-1;
    return x*y;
    x=x+1;
}
```

```

int main()
{
int n=3,m=6;
cout<<f1(f1(m,n),m);
cout<<" "<<m;
}

```

```

int main()
{
int n=3,m=6;
printf("%d ",
f1(f1(m,&n),&m));
printf("%d", m);
}

```

Limbajul Pascal

```

function f1(x:integer; var y:integer):integer;
begin
  x:=x+2;
  y:=y-1;
  f1:=x+y;
  x:=x+1
end;
var m,n: integer;
begin
m:=6;
n:=3;
write(f1(f1(m,n),m), ' ',m)
end.

```

a) 17 5 b) 17 6 c) 10 5 d) 10 6 e) 11 6 f) 10 7

14. Valorile care vor fi memorate în tabloul bidimensional **b**, cu liniile și coloanele numerotate de la 1 la **n**, după apelul **matrice(a,b,n,q)**, unde

$$a = \begin{pmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \end{pmatrix}, \quad b = 0_3, \quad n=3, \quad q=2, \text{ sunt:}$$

Limbajul C++/C

```

void matrice(int a[][100], int b[][100], int n, int q)
{int i,j,k;
  if(q>1)
  {
    for(i=1;i<=n;i++)
      for(j=1;j<=n;j++)
        for(k=1;k<=n;k++)
          b[i][j]=b[i][j]+a[i][k]*a[k][j];
    matrice(a,b,n,q-1);
  }
}

```

Limbajul Pascal

```

type matrix = array [1..100,1..100] of integer;
procedure matrice(a:matrix; b:matrix; n:integer;
q:integer);
var i,j,k: integer;

```



```

begin
  if q>1 then
    begin
      for i:=1 to n do
        for j:=1 to n do
          for k:=1 to n do
            b[i,j]:=b[i,j]+a[i,k]*a[k,j];
          matrice(a,b,n,q-1)
        end
      end;
    end;
  end;

```

$$a) b = \begin{pmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \end{pmatrix}$$

$$c) b = \begin{pmatrix} 4 & 6 & 8 \\ 6 & 8 & 10 \\ 8 & 10 & 12 \end{pmatrix}$$

$$e) b = \begin{pmatrix} 29 & 38 & 47 \\ 38 & 50 & 62 \\ 47 & 62 & 77 \end{pmatrix}$$

$$b) b = \begin{pmatrix} 47 & 38 & 29 \\ 62 & 50 & 38 \\ 77 & 62 & 47 \end{pmatrix}$$

$$d) b = \begin{pmatrix} 4 & 9 & 16 \\ 9 & 16 & 25 \\ 16 & 25 & 36 \end{pmatrix}$$

$$f) b = \begin{pmatrix} 77 & 62 & 47 \\ 62 & 50 & 38 \\ 47 & 38 & 29 \end{pmatrix}$$

15. În urma executării programului de mai jos se afișează:

Limbajul C++

```

#include <iostream>
using namespace std;
void functie(int &a,int &b)
{ b=3*b;
  a=2*a;}
int main()
{ int n=4;
  functie(n,n);
  cout<<n;}

```

Limbajul Pascal

```

procedure functie (var a:integer; var b:integer);
begin
  b:=3*b;
  a:=2*a
end;

var n:integer;
begin
  n:=4;
  functie(n,n);
  write(n)
end.

```

Limbajul C

```

#include <stdio.h>
void functie(int *a,int *b)
{ *b=3*(*b);
  *a=2*(*a);}
int main()
{ int n=4;
  functie(&n,&n);
  printf("%d", n);
}

```

- a) eroare b) 4 c) 6 d) 12 e) 14 f) 24

Varianta 27

1. Rezultatul expresiei de mai jos este:

Limbajul C++/C

`16 / (-5 % 3) * 3`

a) **48** b) **-24** c) **-48**

Limbajul Pascal

`16 div (-5 mod 3) * 3`

d) **-2** e) **2** f) **6**

2. Expresia corespunzătoare penultimei cifre a numărului natural având cel puțin două cifre reținut de variabila întreagă **n** este:

Limbajul C++/C

a) `n / 10 / 10`

c) `n % 10 % 10`

e) `n % 10 / 100`

Limbajul Pascal

a) `n div 10 div 10`

c) `n mod 10 mod 10`

e) `n mod 10 div 100`

b) `n / 10 % 10`

d) `n % 10 / 10`

f) `n / 100 % 10`

b) `n div 10 mod 10`

d) `n mod 10 div 10`

f) `n div 100 mod 10`

3. Afirmatia adevărată în privința secvenței de instrucțiuni de mai jos este:

Limbajul C++/C

`d = 1;`

`while (d * d <= n)`

`{`

`if (n % d == 0)`

`{`

`d1 = d;`

`d2 = n / d;`

`}`

`d++;`

`}`

Limbajul Pascal

`d := 1;`

`while d * d <= n do`

`begin`

`if n mod d = 0 then`

`begin`

`d1 := d;`

`d2 := n div d`

`end;`

`d := d + 1`

`end`

- a) La final **d1** și **d2** vor fi egale **doar** dacă **n** reține un număr prim.
 b) La final **d1** și **d2** vor fi egale **doar** dacă **n** reține cubul unui număr prim.
 c) La final **d1** și **d2** vor fi egale **doar** dacă **n** reține un număr impar.
 d) La final **d1** și **d2** vor fi egale **doar** dacă **n** reține un număr par.
 e) La final **d1** și **d2** vor fi egale **doar** dacă **n** reține 0.
 f) La final **d1** și **d2** vor fi egale **doar** dacă **n** reține un număr pătrat perfect.

4. Secvența de instrucțiuni de mai jos ordonează crescător cele **n** elemente ale tabloului unidimensional **v**, în care primul element este memorat pe poziția 0, dacă punctele de suspensie sunt înlocuite cu:

Limbajul C++/C

`for (i = 0; i < n - 1; i++)`

`{`

`...`

`{`

`if (v[j] > v[j+1])`

Limbajul Pascal

`for i := 0 to n - 2 do`

`begin`

`...`

`begin`

`if v[j] > v[j+1] then`

<pre> { aux = v[j]; v[j] = v[j+1]; v[j+1] = aux; } } </pre>	<pre> begin aux := v[j]; v[j] := v[j+1]; v[j+1] := aux end end </pre>
---	---

Limbajul C++/C

- | | |
|-------------------------------|-------------------------------|
| a) for (j = n-2; j >= i; j--) | b) for (j = 0; j <= i; j++) |
| c) for (j = n-i; j >= i; j--) | d) for (j = 1; j < i; j++) |
| e) for (j = n-1; j > i; j--) | f) for (j = 1; j <= i+1; j++) |

Limbajul Pascal

- | | |
|-------------------------------|-------------------------|
| a) for j := n-2 downto i do | b) for j := 0 to i do |
| c) for j := n-i downto i do | d) for j := 1 to i-1 do |
| e) for j := n-1 downto i+1 do | f) for j := 1 to i+1 do |

5. Subprogramul **f** este definit mai jos. O condiție necesară și suficientă pentru ca numărul natural mai mare strict ca 1 reținut de variabila **n** să fie prim este:

Limbajul C++/C

```

int f(int d, int n)
{
    do
    {
        d++;
    }
    while (n % d != 0);
    return d;
}

```

Limbajul Pascal

```

function f(d, n: integer):
integer;
begin
    repeat
        d := d + 1;
    until n mod d = 0;
    f := d
end;

```

Limbajul C/C++

- | | |
|---------------------|---------------------|
| a) f(2, n) == n | b) f(2, n) == 2 |
| c) f(1, n) == n | d) f(1, n) == 1 |
| e) f(1, n - 1) == n | f) f(2, n - 1) == 2 |

Limbajul Pascal

- | | |
|--------------------|--------------------|
| a) f(2, n) = n | b) f(2, n) = 2 |
| c) f(1, n) = n | d) f(1, n) = 1 |
| e) f(1, n - 1) = n | f) f(2, n - 1) = 2 |

6. Numărul de muchii care trebuie adăugate unui **arbore** cu 10 vârfuri astfel încât acesta să devină **graf complet** este:

- | | | | | | |
|------|-------|-------|-------|-------|-------|
| a) 9 | b) 10 | c) 11 | d) 35 | e) 36 | f) 37 |
|------|-------|-------|-------|-------|-------|

7. Suma elementelor aflate pe diagonala principală a matricei **a**, cu 5 linii și 5 coloane numerotate de la 0 la 4, ale cărei elemente sunt actualizate în secvența de instrucțiuni de mai jos este:

Limbajul C++/C

```

n = 5;
for (i = 0; i < n; i++)

```

Limbajul Pascal

```

n := 5;
for i := 0 to n - 1 do

```

```

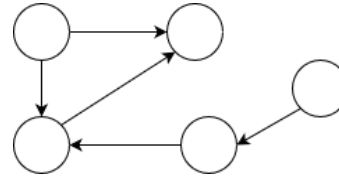
{
    for (j = 0; j < n; j++)
    {
        a[i][j] = (n - i) * n - j;
    }
}

begin
    for j := 0 to n - 1 do
        begin
            a[i,j] := (n-i)*n - j
        end
    end
end

```

a) 15 b) 20 c) 35 d) 55 e) 65 f) 70

8. O variantă care poate corespunde șirului gradelor interne ale vârfurilor grafului orientat alăturat este:



a) (2, 1, 1, 1, 0) b) (1, 1, 1, 1, 0)
c) (2, 1, 0, 2, 0) d) (2, 0, 2, 2, 0)
e) (2, 0, 0, 3, 0) f) (2, 0, 1, 1, 0)

9. Un algoritm Backtracking generează **ultimele două** soluții **pilo** și **poli**, având ca date de intrare cuvântul **poli**. O variantă care poate reprezenta descrierea algoritmului este:

a) Algoritmul generează în ordine invers lexicografică anagramele cuvântului citit.
b) Algoritmul generează în ordine lexicografică anagramele cuvântului citit.
c) Algoritmul generează în ordine lexicografică anagramele cuvântului citit care nu au vocale pe poziții alăturate.
d) Algoritmul generează în ordine lexicografică anagramele cuvântului citit care nu au consoane pe poziții alăturate.
e) Algoritmul generează în ordine lexicografică anagramele cuvântului citit care nu au vocale pe ultima poziție.
f) Algoritmul generează în ordine invers lexicografică anagramele cuvântului citit care nu au consoane pe ultima poziție.

10. Programul de mai jos afișează pe ecran textul **Pol i 2020** dacă punctele de suspensie sunt înlocuite cu:

Limbajul C++/C

```

#include <stdio.h>
#include <string.h>
int main()
{
    char s[256], t[256];
    strcpy(s, "Politehnica 2020");
    ...
    strcpy(s + 4, t);
    puts(s);
    return 0;
}

```

Limbajul C++/C

a) `strcpy(t, strchr(s, ' '));`
c) `strcat(t, strchr(s, '2'));`

Limbajul Pascal

```

var s, t: string;

begin
    s:='Politehnica 2020';
    ...
    s:=copy(s, 1, 4) + t;
    writeln(s)
end.

```

b) `strcpy(t, strcpy(s, ' '));`
d) `strcpy(t, strchr(s, " "));`

e) `strcat(t, strcpy(s, "2"));` f) `strcpy(t, strchr(s, "2"));`

Limbajul Pascal

a) `t:=copy(s, pos(' ', s), 5);` b) `t:=copy(s, copy(' ', s), 4);`

c) `t:=s + pos('2', s);` d) `t:=copy(s, pos(" ", s), 5);`

e) `t:=copy(s, copy("2", s), 4);` f) `t:=copy(s, pos("2", s), 5);`

11. Subprogramul **f** este definit mai jos. Valoarea returnată la apelul **f(24,34)** este:

Limbajul C++/C

```
int f(int a, int b)
{
    int r;
    if (a >= b)
    {
        r = a;
    }
    else if (a % 10 == b % 10)
    {
        r = 2 + f(a + 1, b);
    }
    else if (a % 3 == b % 3)
    {
        r = 1 + f(a + 1, b - 1);
    }
    else
    {
        r = f(a, b - 2);
    }
    return r;
}
```

Limbajul Pascal

```
function f(a, b: integer):
integer;
var r: integer;
begin
    if a >= b then
        begin
            r := a;
        end
    else
        if a mod 10 = b mod 10 then
            begin
                r := 2 + f(a + 1, b);
            end
        else if a mod 3 = b mod 3
        then
            begin
                r := 1 + f(a + 1, b - 1);
            end
        else
            begin
                r := f(a, b - 2);
            end;
        f := r
    end;
end;
```

a) 30

b) 31

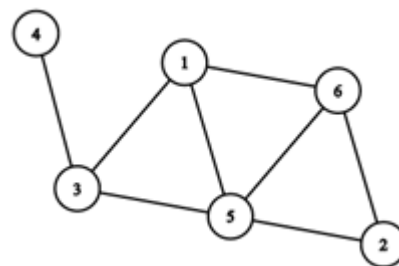
c) 32

d) 33

e) 34

f) 35

12. Numărul maxim de muchii care pot fi eliminate din graful neorientat alăturat astfel încât acesta să conțină cel puțin trei cicluri elementare distincte este:



a) 1

b) 6

c) 2

d) 4

e) 5

f) 3

13. Se generează în ordine lexicografică vectorii de tați corespunzători tuturor arborilor cu rădăcină având exact **6** noduri. Prin înălțimea unui arbore cu rădăcină înțelegem numărul de muchii ale celui mai lung lanț elementar care unește rădăcina cu un alt nod. **A doua** soluție corespunzătoare unui arbore cu înălțimea **3** este:

a) 0 1 2 3 1 1
 c) 0 1 1 1 2 6
 e) 0 1 1 1 1 2

b) 0 1 1 1 2 5
 d) 0 1 1 1 3 5
 f) 0 1 1 1 4 5

14. Subprogramul **rad** de mai jos calculează și returnează cel mai mic număr care ridicat la pătrat este mai mare sau egal cu numărul natural reținut de **x** (partea întreagă superioară a lui radical din **x**) dacă punctele de suspensie sunt înlocuite cu:

Limbajul C++/C

```
int rad(int s, int d, int x)
{
    int rez, m;
    if (s == d)
    {
        rez = s;
    }
    else
    {
        m = (s + d) / 2;
        if (...)
        {
            rez = rad(s, m, x);
        }
        else
        {
            rez = rad(m + 1, d, x);
        }
    }
    return rez;
}
```

Limbajul Pascal

```
function rad(s,d,x: integer)
:integer;
var m, rez: integer;
begin
    if s = d then
        begin
            rez := s
        end
    else
        begin
            m := (s + d) div 2;
            if ... then
                begin
                    rez := rad(s, m, x)
                end
            else
                begin
                    rez := rad(m+1,d, x)
                end
            end;
        rad := rez
    end;
```

Limbajul C++/C

a) **m * m == x**
 c) **m * m <= x**
 e) **m * m < x**

b) **m * m >= x**
 d) **m * m > x**
 f) **m * m != x**

Limbajul Pascal

a) **m * m = x**
 c) **m * m <= x**
 e) **m * m < x**

b) **m * m >= x**
 d) **m * m > x**
 f) **m * m <> x**

15. Fie un tablou unidimensional **v** care reține **n** numere naturale: **v[0]**, **v[1]**, ..., **v[n-1]** și un număr întreg **t**. Secvența de instrucțiuni de mai jos are ca efect obținerea lungimii maxime **lmax** a unei subsecvențe **v[k]**, **v[k+1]**, ... **v[k+lmax-1]** având suma elementelor mai mică sau egală cu **t** dacă punctele de suspensie sunt înlocuite cu:

Limbajul C++/C

```
s = 0;
j = 0;
lmax = 0;
```

Limbajul Pascal

```
s := 0;
j := 0;
lmax := 0;
```

```

for (i = 0; i < n; i++)
{
    s += v[i];
    while (j <= i && s > t)
    {
        ...
        j++;
    }
    if (i - j + 1 > lmax)
    {
        lmax = i - j + 1;
    }
}

```

```

for i := 0 to n-1 do
begin
    s := s + v[i];
    while(j<=i) and (s>t) do
        begin
            ...
            j := j + 1
        end;
    if i - j + 1 > lmax then
        begin
            lmax := i - j + 1
        end;
    end;
end;

```

Limbajul C++/C

- a) **s += v[j];**
- c) **s -= v[i];**
- e) **s += v[i];**

Limbajul Pascal

- a) **s := s+v[j];**
- c) **s := s-v[i];**
- e) **s := s+v[i];**

- b) **i--;**
- d) **s -= v[j];**
- f) **i++;**

- b) **i := i-1;**
- d) **s := s-v[j];**
- f) **i := i+1;**

Varianta 28

1. Expresia corespunzătoare mediei aritmetice a patru numere reale memorate în variabilele **a, b, c** și **d** este:

- | | |
|----------------------|-----------------------|
| a) $a+b+c+d/4$ | b) $(a+b+c+d) * 1/2$ |
| c) $(a+b+c+d) * 0.4$ | d) $(a+b+c+d) * 0.25$ |
| e) $(a+b+c+d) * 4.0$ | f) $(a+b+c+d) * 1.4$ |

2. În secvențele de instrucțiuni **S1** și **S2** variabilele **n** și **p** sunt de tip întreg. Obținerea în variabila **p** a primei cifre a numărului reținut inițial de **n** este realizată:

Limbajul C/C++

```
//S1
p = n;
while (p > 9)
{
    p /= 10;
}
```

//S2

```
do
{
    p = n % 10;
    n /= 10;
}
while (n != 0);
```

Limbajul Pascal

```
{S1}
p := n;
while p > 9 do
begin
    p := p div 10
end
```

{S2}

```
repeat
    p := n mod 10;
    n := n div 10
until n = 0;
```

- | | |
|--|--|
| a) doar de S1 | b) doar de S2 |
| c) atât de S1 , cât și de S2 | d) doar folosind o cu totul altă secvență |
| e) doar de S1 , dacă n are o singură cifră | f) doar de S2 , dacă n are mai multe cifre |

3. În urma executării secvenței de instrucțiuni de mai jos variabila **nr** va reține numărul divizorilor primi ai numărului natural nenul reținut inițial de variabila **n** dacă punctele de suspensie sunt înlocuite cu:

Limbajul C++/C

```
d = 2;
nr = 0;
while (n > 1)
{
    p = 0;
    while (...)
    {
        p = 1;
        n /= d;
    }
    nr += p;
    d++;
}
```

Limbajul Pascal

```
d := 2;
nr := 0;
while n > 1 do
begin
    p := 0;
    while ... do
begin
        p := 1;
        n := n div d
    end;
    nr := nr + p;
    d := d + 1
end
```


Limbajul C++/C

a) $d*d < n$ b) $n < d$ c) $n > d$ d) $d+d < n$ e) $n \% d != 0$ f) $n \% d == 0$

Limbajul Pascal

a) $d*d < n$ b) $n < d$ c) $n > d$ d) $d+d < n$ e) $n \bmod d < > 0$ f) $n \bmod d = 0$

4. Vectorul $\mathbf{v}$ are n componente întregi, numerotate de la 0, ordonate crescător. Pentru ca în urma executării secvenței de instrucțiuni de mai jos să se insereze valoarea întreagă reținută de $\mathbf{x}$ în vectorul $\mathbf{v}$ și acesta să rămână ordonat, punctele de suspensie trebuie înlocuite cu:

Limbajul C++/C

```
i = n - 1;
while (i >= 0 && v[i] > x)
{
    v[i+1] = v[i];
    i--;
}
...;
n++;
```

Limbajul C++/C

a) $\mathbf{x} = \mathbf{v}[\mathbf{i}+1]$
c) $\mathbf{v}[\mathbf{n}] = \mathbf{x}$
e) $\mathbf{v}[\mathbf{i}+1] = \mathbf{x}$

Limbajul Pascal

a) $\mathbf{x} := \mathbf{v}[\mathbf{i}+1]$
c) $\mathbf{v}[\mathbf{n}] := \mathbf{x}$
e) $\mathbf{v}[\mathbf{i}+1] := \mathbf{x}$

Limbajul Pascal

```
i := n - 1;
while (i>=0) and (v[i]>x) do
begin
    v[i+1] := v[i];
    i := i - 1;
end;
...;
n := n + 1;
```

b) $\mathbf{v}[\mathbf{i}] = \mathbf{x}$
d) $\mathbf{v}[\mathbf{i}-1] = \mathbf{x}$
f) $\mathbf{v}[\mathbf{n}+1] = \mathbf{x}$

b) $\mathbf{v}[\mathbf{i}] := \mathbf{x}$
d) $\mathbf{v}[\mathbf{i}-1] := \mathbf{x}$
f) $\mathbf{v}[\mathbf{n}+1] := \mathbf{x}$

5. În urma executării secvenței de instrucțiuni de mai jos suma elementelor pare ale matricei $\mathbf{a}$, cu 5 linii și 5 coloane numerotate de la 0 la 4 va fi:

Limbajul C/C++

```
n = 5;
for (i = 0; i < n; i++)
{
    for (j = 0; j < n; j++)
    {
        a[i][j] = i - j + n;
    }
}
```

a) 20 b) 60 c) 62 d) 64 e) 61 f) 12

Limbajul Pascal

```
n := 5;
for i := 0 to n - 1 do
begin
    for j := 0 to n - 1 do
begin
    a[i,j] := i - j + n;
end;
end;
```

6. Subprogramul $\mathbf{f}$ este definit mai jos. Apelul care returnează valoarea 0 este:

Limbajul C++/C

```
int f(int n)
{
    int r = 0;
    while (r * r < n)
    {
        r++;
    }
}
```

Limbajul Pascal

```
function f(n:integer):integer;
var r: integer;
begin
    r := 0;
    while r * r < n do
begin
```

```

    }
    return r * r - n;
}
                                r := r + 1
                                end;
                                f := r*r - n
                                end;

```

a) **f(23)** b) **f(225)** c) **f(17)** d) **f(131)** e) **f(122)** f) **f(1000)**

7. Un graf orientat tare conex are șirul gradelor externe ale vârfurilor sale **(3,1,1,1)**. Graful nu are arce cu extremitățile identice (bucle). O variantă care poate reprezenta șirul gradelor interne ale vârfurilor grafului este:

a) **(6,0,0,0)** b) **(2,1,3,2)**
c) **(2,2,2,0)** d) **(1,1,2,2)**
e) **(1,1,1,2)** f) **(0,0,1,5)**

8. Numărul nodurilor terminale (frunze) ale arborelui cu rădăcină corespunzător vectorului de tați **(7,4,0,3,7,3,3,4)** este:

a) **3** b) **7** c) **5** d) **0** e) **8** f) **4**

9. Programul de mai jos afișează numărul aparițiilor caracterului **c** în cuvântul **s** dacă punctele de suspensie sunt înlocuite cu:

Limbajul C++/C

```

#include <stdio.h>
#include <string.h>

int main()
{
    char s[256], c, *p;
    int nr;
    scanf("%s %c", s, &c);
    p = strchr(s, c);
    nr = 0;
    while (p != NULL)
    {
        nr++;
        ...;
        p = strchr(s, c);
    }
    printf("%d", nr);
    return 0;
}

```

}

Limbajul C/C++

a) **strcpy(p,p+1)**
c) **strcat(s,p+1)**
e) **strcat(s,p)**

Limbajul Pascal

a) **delete(s,p,1)**
c) **concat(s,p+1,1)**
e) **concat(s,p,1)**

Limbajul Pascal

```

var s: string;
    c: char;
    p, nr: integer;

begin
    readln(s);
    readln(c);
    p := pos(c, s);
    nr := 0;
    while p <> 0 do
        begin
            nr := nr + 1;
            ...;
            p := pos(c, s);
        end;
    writeln(nr);
end.

```

b) **strcpy(p+1,p)**
d) **p++**
f) **p--**

b) **delete(p,s,1)**
d) **p := p + 1**
f) **p := p - 1**

- 10 Apelul **s (3)** al subprogramului **s** definit mai jos va afișa pe ecran:
- | | |
|--|--|
| <p>· Limbajul C++/C</p> <pre>void s(int n) { int i; if (n > 0) { for (i = 0; i < n; i++) { printf("%d", i); s(i - 1); } printf("%d", n); } }</pre> | <p>Limbajul Pascal</p> <pre>procedure s(n: integer); var i: integer; begin if n > 0 then begin for i := 0 to n-1 do begin write(i); s(i-1) end; write(n) end end;</pre> |
|--|--|
- a) 012301 b) 012031
c) 00120013 d) 01203
e) 0120013 f) 012013
- 11 Rezolvarea problemei generării tuturor imaginilor funcțiilor injective definite pe mulțimea $\{1, 2, \dots, k\}$ cu valori în mulțimea $\{1, 2, \dots, n\}$ prin metoda Backtracking necesită ca fiecare element adăugat în vectorul soluție să respecte o condiție de compatibilitate cu cele deja introduse. Aceeași condiție este respectată în cazul:
- a) Generării tuturor submulțimilor mulțimii $\{1, 2, \dots, n\}$.
b) Generării tuturor permutărilor mulțimii $\{1, 2, \dots, n\}$.
c) Generării produsului cartezian a k mulțimi $\{1, 2, \dots, n\}$.
d) Generării submulțimilor având k elemente ale mulțimii $\{1, 2, \dots, n\}$.
e) Generării tuturor partițiilor mulțimii $\{1, 2, \dots, n\}$.
f) Generării submulțimilor având cel puțin k elemente ale mulțimii $\{1, 2, \dots, n\}$.
- 12 Fie **G** un graf neorientat cu **10** vârfuri și **8** muchii. Afirmatia **falsă** este:
- a) **G** nu poate fi conex
b) **G** poate avea mai multe cicluri elementare
c) **G** nu poate fi hamiltonian
d) **G** nu poate fi eulerian
e) **G** poate avea vârfuri izolate (de grad 0)
f) **G** poate avea vârfuri terminale (de grad 1)
- 13 Un arbore cu rădăcină cu **12** noduri are proprietatea că exact **3** dintre nodurile sale au câte **3** fii. Prin înălțimea unui arbore cu rădăcină înțelegem numărul de muchii ale celui mai lung lanț care unește rădăcina cu un alt nod. Înălțimea maximă a arborelui este:
- a) 1 b) 2 c) 3 d) 4 e) 5 f) 6
- 14 Subprogramul **mysort** de mai jos ordonează crescător componentele întregi ale vectorului **v** (declarat astfel încât să poată reține cel mult **100** de elemente, numerotate de la **0** la **n-1**) dacă punctele de suspensie sunt înlocuite cu:

Limbajul C++/C

```
void mysort(int n,int
v[100])
{
    int aux;
    if (n > 1)
    {
        mysort(n - 1, v);
        ...
    }
}
```

Limbajul C++/C

```
a)
if (v[n-2] > v[n-1])
{
    aux = v[n-1];
    v[n-1] = v[n-2];
    v[n-2] = aux;
    mysort(n - 1, v);
}
c)
int i;
for (i=0; i+1<n; i++)
{
    if (v[i] > v[i+1])
    {
        aux = v[i];
        v[i] = v[i+1];
        v[i+1] = aux;
    }
}
e)
if (v[n] > v[n+1])
{
    aux = v[n-1];
    v[n-1] = v[n-2];
    v[n-2] = aux;
    mysort(n - 1, v);
}
```

Limbajul Pascal

```
a)
if v[n-2] > v[n-1] then
begin
    aux := v[n-1];
    v[n-1] := v[n-2];
    v[n-2] := aux;
```

Limbajul Pascal

```
procedure mysort
(n:integer; var v:vector);
var aux:integer;
begin
    if n>1 then
    begin
        mysort(n - 1, v);
        ...
    end
end;
```

Observație: Tipul **vector** a fost declarat anterior:

```
type vector = array [0..99]
of integer;
```

```
b)
if (v[n-2] < v[n-1])
{
    aux = v[n-1];
    v[n-1] = v[n-2];
    v[n-2] = aux;
    mysort(n - 1, v);
}
d)
int i;
for (i=n-1; i-1>=0; i--)
{
    if (v[i] > v[i-1])
    {
        aux = v[i];
        v[i] = v[i-1];
        v[i-1] = aux;
    }
}
f)
if (v[n+1] < v[n])
{
    aux = v[n-1];
    v[n-1] = v[n-2];
    v[n-2] = aux;
    mysort(n - 1, v);
}
```

```
b)
if v[n-2] < v[n-1] then
begin
    aux := v[n-1];
    v[n-1] := v[n-2];
    v[n-2] := aux;
```

```

        mysort(n-1, v)
    end
c)
for i:=0 to n-2 do
    begin
        if v[i] > v[i+1] then
            begin
                aux := v[i];
                v[i] := v[i+1];
                v[i+1] := aux
            end
        end
    end
e)
if v[n] > v[n+1] then
    begin
        aux := v[n-1];
        v[n-1] := v[n-2];
        v[n-2] := aux;
        mysort(n-1, v)
    end
end

```

```

        mysort(n-1, v)
    end
d)
for i:=n-1 downto 0 do
    begin
        if v[i] > v[i-1] then
            begin
                aux := v[i];
                v[i] := v[i-1];
                v[i-1] := aux
            end
        end
    end
f)
if v[n+1] < v[n] then
    begin
        aux := v[n-1];
        v[n-1] := v[n-2];
        v[n-2] := aux;
        mysort(n-1, v)
    end
end

```

- 15 Fie un vector $\mathbf{v}$ care reține cele n cifre ($\mathbf{v}[0], \mathbf{v}[1], \dots, \mathbf{v}[n-1]$) ale unui număr natural $\mathbf{X}$ și un număr natural $\mathbf{k}$ ($\mathbf{k} < \mathbf{n}$). Secvența de instrucțiuni de mai jos își propune să construiască vectorul $\mathbf{s}$, care să rețină cifrele celui mai mare număr natural $\mathbf{Y}$ care poate fi obținut din $\mathbf{X}$ prin eliminarea a exact $\mathbf{k}$ cifre, fără a schimba ordinea în care cifrele apăreau în $\mathbf{X}$. De exemplu, dacă $\mathbf{n}=8, \mathbf{k}=3, \mathbf{v}=(5, 1, 3, 5, 4, 4, 6, 9)$, corespunzător lui $\mathbf{X}=51354469$, secvența de cod ar trebui să construiască $\mathbf{s}=(5, 5, 4, 6, 9)$, corespunzător lui $\mathbf{Y}=55469$. Pentru a obține rezultatul dorit punctele de suspensie trebuie înlocuite cu:

Limbajul C++/C

```

m = 0;
for (i = 0; i < n; i++)
{
    while (m>0 && k>0 &&...)
    {
        m--;
        k--;
    }
    s[m++] = v[i];
}
m -= k;

```

- a) $\mathbf{v}[i] > \mathbf{v}[i-1]$
 c) $\mathbf{v}[i] > \mathbf{s}[m-1]$
 e) $\mathbf{v}[i] < \mathbf{s}[m-1]$

Limbajul Pascal

```

m := 0;
for i := 0 to n - 1 do
    begin
        while (m>0) and (k>0) and (...)
        do
            begin
                m := m - 1;
                k := k - 1;
            end;
        s[m] := v[i];
        m := m + 1
    end;
m := m - k;

```

- b) $\mathbf{v}[i] > \mathbf{s}[m-1]$
 d) $\mathbf{v}[i] < \mathbf{v}[i-1]$
 f) $\mathbf{v}[i] < \mathbf{v}[i+1]$

Varianta 29

1. Răsturnatul tabloului unidimensional (2 4 1 3 7 0 5) este (5 0 7 3 1 4 2).
Numărul necesar de interschimbări pentru a răsturna un tablou unidimensional cu n (număr natural nenul, *impar*) elemente este:
a) 1 b) $n/2+1$ c) $(n-1)/2$
d) $(n+1)/2$ e) $n/2-1$ f) n
2. După permutarea circulară spre stânga cu 2 poziții, tabloul unidimensional (18 91 1 15 102) devine:
a) (102 15 18 91 1) b) (1 15 102 18 91) c) (1 15 102 91 18)
d) (15 102 18 91 1) e) (91 1 15 102 18) f) (1 15 102)
3. În șirurile de mai jos, elementul de pe poziția k reprezintă rândul pe care este așezată a k -a damă (regină) pe o tablă de șah, damele fiind așezate pe coloane distincte (dama 1 pe coloana 1, dama 2 pe coloana 2, ș.a.m.d.).
Pentru a așeza 4 dame (regine) pe o tablă de șah 4×4 , astfel încât acestea să nu se atace între ele (două dame se atacă atunci când se află pe aceeași linie, pe aceeași coloană sau pe aceeași diagonală), o soluție corectă este:
a) 4 3 2 1 b) 4 2 3 1 c) 3 1 4 2
d) 2 3 1 4 e) 2 1 3 4 f) 1 2 3 4
4. Pentru a sorta crescător tabloul unidimensional (10 24 9 11 33 7 15), folosind BubbleSort, numărul de interschimbări necesare este:
a) 9 b) 10 c) 11 d) 12 e) 13 f) 14
5. Cu ajutorul metodei backtracking se generează, în ordine crescătoare, numere cu proprietățile:
 - au exact cinci cifre;
 - cifrele de pe poziții consecutive sunt în ordine strict crescătoare;
 - au cel mult două cifre alăturate de aceeași paritate;Exemplu de numere generate: 13469, 14589.
O secvență care conține cinci numere generate consecutiv este:
a) 45678 45679 45689 46789 56789
b) 34789 35678 35679 35689 45678
c) 34578 34569 34568 34567 26789
d) 13458 13459 13467 13478 13479
e) 13458 13459 13467 13468 13469
f) 26789 34567 34568 34569 34578

6. Pentru funcția **f** definită mai jos, valoarea returnată de apelul **f(2019,2347)** ; este:

Limbajul C++/C

```
int f(int a, int b)
{
    int cif;
    if (a+b>0)
    {
        cif=a%10;
        if (cif<b%10)
            cif=b%10;
        return f(a/10, b/10)*10+cif;
    }
    return 0;
}
```

Limbajul Pascal

```
function f(a,b: integer):integer;
var cif:integer;
begin
    if (a+b>0) then
    begin
        cif:=a mod 10;
        if (cif < b mod 10) then
            cif:=b mod 10;
        f:=f(a div 10, b div 10)*10+cif
    end
    else
        f:=0
    end;
end;
```

- | | | |
|---------|---------|---------|
| a) 349 | b) 2017 | c) 2349 |
| d) 7102 | e) 9432 | f) 9743 |

7. Pentru tabloul unidimensional **(4, 6, 14, 25, 61, 73, 82, 87, 95, 96, 98)** numărul minim de elemente ale tabloului care trebuie verificate până este găsit elementul **82** este:

- | | | | | | |
|------|------|------|------|------|------|
| a) 7 | b) 6 | c) 5 | d) 3 | e) 2 | f) 1 |
|------|------|------|------|------|------|

8. În urma executării programului de mai jos, variabila **k** are valoarea:

Limbajul C++/C

```
#include <iostream>
int k=1;
int f(int n)
{
    int k;
```

Limbajul Pascal

```
program p;
function f(n: integer):integer;
var k:integer;
begin
    k:=k+2;
```

<pre> k=k+2; return k; } int main() { k=f(k); return 0; } </pre>	<pre> f:=k end; var k:integer; begin k:=1; k:=f(k) end. </pre>
--	--

- | | | |
|------|---------------|---------------------------------------|
| a) 0 | b) 1 | c) 2 |
| d) 3 | e) nedefinită | f) nicio valoare, programul are erori |

9. Numărul elementelor care se găsesc strict deasupra diagonalei secundare a unui tablou bidimensional cu **20** de linii și **20** de coloane este:

- | | | |
|--------|--------|--------|
| a) 180 | b) 190 | c) 200 |
| d) 210 | e) 380 | f) 400 |

10. Problema *Turnurile din Hanoi*:

Se dau 3 tije. Pe prima tijă se găsesc discuri de diametre diferite, așezate în ordinea descrescătoare a diametrelor privite de jos în sus. Se cere să se mute discurile de pe prima tijă pe cea de-a doua, utilizând ca tijă intermediară cea de-a treia, respectând următoarele reguli:

- la fiecare pas se mută un singur disc;
- nu este permis să se așeze un disc cu diametrul mai mare peste un disc cu diametrul mai mic.

Numărul minim de mutări necesare rezolvării problemei *Turnurile din Hanoi* pentru **10** discuri este:

- | | | |
|---------|---------|---------|
| a) 99 | b) 100 | c) 1022 |
| d) 1023 | e) 1024 | f) 1025 |

11. Într-un graf orientat cu **56** de arce, în care oricare arc are extremități distincte și oricare două arce diferă prin cel puțin una dintre extremități, numărul minim de vârfuri este:

- | | | |
|-------|-------|--------|
| a) 6 | b) 7 | c) 8 |
| d) 28 | e) 56 | f) 112 |

12. Fie problema:

Se dau **$n-1$** numere naturale distincte de la **1** la **n** ($1 < n < 10^5$). Se cere un algoritm care să determine numărul lipsă.

Fie algoritmii:

A₁: Se verifică prin câte o parcurgere prezența fiecărui număr de la **1** la **n** în șir.

A₂: Numărul lipsă este egal cu diferența dintre $[n \cdot (n + 1) / 2]$ și suma numerelor din șir.

A₃: Se sortează numerele și se determină pentru ce valori consecutive în șirul sortat diferența este diferită de 1.

A₄: Se sortează crescător numerele și se determină prima valoare din șirul sortat care este diferită de poziția în șir.

Este adevărat enunțul:

- a) Algoritmii A₁ și A₂ rezolvă problema pentru anumite date de intrare.
- b) Algoritmul A₂ este cel mai puțin eficient din punctul de vedere al timpului de executare.
- c) Algoritmul A₄ este cel mai eficient din punctul de vedere al timpului de executare.
- d) Algoritmul A₄ rezolvă problema doar dacă numărul lipsă este cel mai mare din șir.
- e) Cel puțin unul dintre algoritmi nu rezolvă problema.
- f) Doi dintre algoritmi nu diferă ca eficiență din punctul de vedere al timpului de executare.

13. Fie enunțurile:

E₁: orice graf neorientat conex **G** cu cel puțin 2 noduri, conține cel puțin un nod **k** care poate fi eliminat (și muchiile incidente cu el) obținându-se un subgraf **G'** conex;

E₂: un graf neorientat cu **n** (**n**>2) noduri și **n** muchii conține cel puțin un ciclu;

E₃: orice arbore cu **n** (**n**>1) noduri conține cel puțin două noduri cu gradul 1.

Enunțurile adevărate sunt:

- | | | |
|--|--|--|
| a) doar E ₁ | b) doar E ₂ | c) doar E ₁ și E ₂ |
| d) doar E ₁ și E ₃ | e) doar E ₂ și E ₃ | f) E ₁ , E ₂ și E ₃ |

14. În urma executării unui program pentru generarea permutărilor elementelor unui șir de caractere ce conține duplicate, numărul de cuvinte distincte, anagrame ale cuvântului “**caracter**”, este:

- | | | |
|----------|----------|----------|
| a) 120 | b) 2520 | c) 5040 |
| d) 10080 | e) 20160 | f) 40320 |

15. Fie următoarele formule:

$$1. F(n) = \begin{cases} \left[2 \cdot F\left(\frac{n}{2} - 1\right) + F\left(\frac{n}{2}\right) \right] \cdot F\left(\frac{n}{2}\right), & n \text{ este par}, n > 3 \\ \left[F\left(\frac{n+1}{2}\right) \right]^2 + \left[F\left(\frac{n-1}{2}\right) \right]^2, & n \text{ este impar}, n > 2 \end{cases}$$

$$2. F(n) = \frac{1}{\sqrt{5}} \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n + \frac{1}{\sqrt{5}} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n$$

$$3. F(n) = \frac{1}{\sqrt{5}} \cdot \left(\frac{1+\sqrt{5}}{2}\right)^{n/2} + \frac{1}{\sqrt{5}} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^{n/2}$$

Știind că $F(1) = 1, F(2) = 1$, pentru a determina al **n**-lea (**n**>2) termen din șirul lui Fibonacci (1, 1, 2, 3, 5, 8, ...) se poate folosi:

- | | |
|-------------------------------------|--------------------------------|
| a) niciuna dintre cele trei formule | b) doar formula 1 |
| c) doar formula 2 | d) doar formula 1 și formula 2 |
| e) doar formula 3 | f) toate cele trei formule |

Varianta 30

- Răsturnatul tabloului unidimensional (2 4 1 3 7 0) este (0 7 3 1 4 2).
Numărul necesar de interschimbări pentru a răsturna un tablou unidimensional cu n (număr natural nenul, par) elemente este:
a) 1 b) $n/2-1$ c) $n/2$ d) $(n-1)/2$ e) $n/2+1$ f) n
- Pentru a permuta, eficient din punctul de vedere al memoriei utilizate, circular spre dreapta cu k poziții elementele unui tablou unidimensional cu n numere întregi (n, k numere naturale nenule, $k \leq n$) este necesar un spațiu suplimentar de memorie de:
a) $n \cdot k$ elemente b) n elemente c) k elemente
d) 0 elemente e) $2 \cdot n$ elemente f) $2 \cdot k$ elemente
- Numărul soluțiilor de așezare a 3 dame (regine) pe o tablă de șah 3×3 , astfel încât acestea să nu se atace între ele (două dame se atacă atunci când se află pe aceeași linie, pe aceeași coloană sau pe aceeași diagonală), este:
a) 5 b) 4 c) 3 d) 2 e) 1 f) 0
- Un graf este memorat printr-o matrice de adiacență cu $x+5$ linii și $y+3$ coloane. Valorile lui x și respectiv y ar putea fi:
a) 5 3 b) 3 5 c) 1 4 d) 4 1 e) 2 2 f) 2 1
- $dc(a, b)$ reprezintă o funcție care determină cel mai mare divizor comun al numerelor naturale a și b iar $a \bmod b$ reprezintă restul împărțirii numărului întreg a la numărul întreg nenul b .
O formulă recursivă pentru determinarea celui mai mare divizor comun a două numere x și y este:
a) $dc(x, y) = dc(x \cdot y, y)$ b) $dc(x, y) = dc(x \bmod y, x)$
c) $dc(x, y) = dc(y, x \cdot y)$ d) $dc(x, y) = dc(x, x \bmod y)$
e) $dc(x, y) = dc(y, x \bmod y)$ f) $dc(x, y) = dc(x \bmod x, y \bmod y)$
- Fie un tablou unidimensional. Algoritmul de sortare rapidă (quick sort) împarte tabloul în:
a) 2 subtablouri, întotdeauna cu același număr de elemente
b) 2 subtablouri, nu întotdeauna cu același număr de elemente
c) 3 subtablouri, întotdeauna cu același număr de elemente
d) 3 subtablouri, nu întotdeauna cu același număr de elemente
e) 4 subtablouri, întotdeauna cu același număr de elemente

f) 4 subtablouri, nu întotdeauna cu același număr de elemente

7. Fie șirul de caractere **tablou**. Răsturnatul acestui șir este **uolbat**.

Structura de date cea mai adecvată în care se poate memora un șir de caractere pentru a-l folosi răsturnat este:

- a) arbore b) coadă c) graf orientat
d) o coadă și un graf orientat e) stivă f) o coadă și un graf neorientat

8. În programul de mai jos subprogramul **f** este definit incomplet.

Limbajul C++/C

```
void f(int n)
{
    if (n!=0)
    {
        cout<<n; |printf ("%d",n) ;
        .....
    }
}
```

Limbajul Pascal

```
procedure f(n:integer);
begin
    if (n<>0) then
    begin
        writeln(n);
        .....
    end
end;
```

Instrucțiunea cu care se pot înlocui punctele de suspensie astfel încât după apelul **f (n)** din programul principal executarea să se încheie fără niciun fel de eroare, indiferent de valoarea întreagă a parametrului, este:

Limbajul C++/C

- a) **f (n-2) ;** b) **f (n-1) ;**
c) **f (n%2) ;** d) **f (n/2) ;**
e) **f (n+2) ;** f) **f (n*2) ;**

Limbajul Pascal

- a) **f (n-2)** b) **f (n-1)**
c) **f (n mod 2)** d) **f (n div 2)**
e) **f (n+2)** f) **f (n*2)**

9. Subprogramele **f** și **s** sunt definite mai jos.

Limbajul C++/C

```
int f(int x)
{
    x=x+1;
    return x;
}
int s(int x, int y)
{
    return x+y;
}
```

Limbajul Pascal

```
function f(x: integer): integer;
begin
    x:=x+1;
    f:=x
end;
function s(x,y: integer): integer;
begin
    s:=x+y
end;
```

În urma executării instrucțiunii

Limbajul C++/C

```
z=s (f (1) , f (1) ) ;
```

Limbajul Pascal

```
z:=s (f (1) , f (1) ) ;
```

variabila de tip întreg **z** are valoarea:

- a) 0 b) 1 c) 2 d) 3 e) 4 f) 5

10. În mulțimea de numere naturale de la **101** la **200** numărul celor care nu sunt divizibile cu niciuna dintre valorile **2**, **3** și **5** este:
a) **25** b) **26** c) **27** d) **28** e) **29** f) **30**
11. În urma executării unui program pentru generarea permutărilor elementelor unui șir care conține elemente care apar de mai multe ori, rezultatul permutării elementelor șirului de caractere "**xx**", este:
a) **xx** b) **xx, xx** c) **x, x**
d) **x** e) **x, xx** f) **xx, x**
12. Cu ajutorul metodei backtracking se generează, în ordine crescătoare, numere naturale cu proprietățile:
- au exact cinci cifre;
- cifrele de pe poziții consecutive sunt în ordine strict crescătoare;
- au cel mult două cifre alăturate de aceeași paritate;
Exemplu de numere generate: **13469, 14589**.
Fie următoarele enunțuri:
1. se generează cel mult **27** de numere cu prima cifră **2**;
2. se generează exact șase numere de forma *ippii*, unde *i* este o cifră impară iar *p* este o cifră pară;
3. există numere generate care să aibă patru cifre de aceeași paritate;
4. cifrele **2** și **7** nu pot apărea pe poziții consecutive în numerele generate;
5. în numerele generate cifra **1** apare pe prima poziție de exact același număr de ori cum cifra **9** apare pe ultima poziție.
Numărul de enunțuri adevărate este:
a) **0** b) **1** c) **2** d) **3** e) **4** f) **5**
13. Numărul ciclurilor hamiltoniene distincte într-un graf neorientat complet K_n , cu $n \geq 3$ noduri, este:
a) $2^{n(n-1)/2}$ b) $4^{n(n-1)/2}$ c) $(n-1)!$
d) $(n-1)!/2$ e) $(n+1)!/2$ f) $n!/2$
14. Un graf orientat este complet dacă oricare două vârfuri distincte ale sale sunt adiacente. Dacă numărul de grafuri orientate complete ce se pot obține cu **n** vârfuri este **59049**, valoarea lui **n** este:
a) **4** b) **5** c) **6** d) **7** e) **8** f) **9**

Fie următoarele relații:

15. $E_1: F_p(n) = F(3 \cdot n);$

$E_2: F_p(n) = 4 \cdot F_p(n-1) + F_p(n-2), n \geq 2, F_p(0) = 0 \text{ și } F_p(1) = 2.$

$E_3: F_p(n) = F\left(\frac{n+1}{2}\right) \cdot F\left(\frac{n+1}{2}\right) + F_p\left(\frac{n-1}{2}\right) \cdot F_p\left(\frac{n-1}{2}\right), n \geq 1 \text{ și } F_p(0) = 0.$

unde: $F(n)$ este al n -lea termen din șirul lui Fibonacci (**1, 1, 2, 3, 5, 8, ...**), iar $F_p(n)$ este al n -lea termen par din șirul lui Fibonacci (**2, 8, 34, ...**).

Pentru a determina al n -lea termen par din șirul lui Fibonacci putem folosi:

a) doar relația E_1

b) doar relația E_2

c) doar relațiile E_1 și E_2

d) doar relația E_3

e) doar relațiile E_1 și E_3

f) doar relațiile E_2 și E_3

Varianta 31

1. Fie expresia:

Limbajul C++/C

`2020 - n%2020 + n/2020`

Limbajul Pascal

`2020 - n mod 2020 + n div 2020`

Indicați care este valoarea maximă a expresiei de mai sus știind că variabila întreagă **n** memorează un număr natural cu cel mult 4 cifre.

- a) 0 b) 1 c) 2020 d) 2024 e) 2080 f) 4039

2. Variabilele întregi **x** și **y** memorează numere naturale. Precizați ce se afișează după executarea instrucțiunilor de mai jos.

Limbajul C++/C

```
for(x=0; x<=3; x++)
  for(y=3; y>=x; y--)
    if (y%3==2)
      cout<<x+y; | printf("%d",x+y);
```

Limbajul Pascal

```
for x :=0 to 3 do
  for y :=3 downto x do
    if y mod 3 = 2
      then write(x+y);
```

- a) 234 b) 5432 c) 22525 d) 54321 e) 654321 f) 6543210

3. Variabilele **i** și **j** sunt de tip întreg, iar variabila **a** memorează un tablou bidimensional cu 10 linii și 10 coloane, având inițial toate valorile elementelor egale cu zero. Suma valorilor elementelor din tabloul **a**, după executarea instrucțiunilor de mai jos, este:

Limbajul C++/C

```
for(x=1; x<=6; x++)
  for(y=1; y<=6; y++)
    if (x%2==0)
      a[x][y]=(x-1)%5;
      else a[y][x]= y-1;
```

Limbajul Pascal

```
for x :=1 to 6 do
  for y :=1 to 6 do
    if x mod 2 = 0 then
      a[x, y] := (x-1) mod 5
      else a[y, x] := y-1;
```

- a) 80 b) 72 c) 69 d) 55 e) 48 f) 42

4. Șirul de caractere afișat după executarea instrucțiunilor de mai jos este:

Limbajul C++/C

```
char s[20]="BUTONOMATICA";
strcpy(s+5,s+6);
s[0]=s[0]-1;
strcpy(s+5,s+6);
cout<<s; | printf("%s",s);
```

Limbajul Pascal

```
var s: string[20];
s:='BUTONOMATICA';
delete(s,6,1);
s[1]:= chr(ord(s[1])-1);
delete(s,6,1);
write(s);
```

- a) AUTONATICA b) AUTOMATICA c) AUTONTICA
d) AUTOnATIC e) Auton f) butonatica

5. În secvența de instrucțiuni de mai jos, atât variabila **I**, cât și variabila **J** memorează în câmpurile **a** și **b** numere reale reprezentând extremitatea stângă, respectiv extremitatea dreaptă a câte unui interval deschis de numere reale (**a, b**), unde **a < b**.

Limbajul C++ struct interval { float a,b;}; interval I,J;	Limbajul C typedef struct { float a,b; }interval; interval I,J;	Limbajul Pascal type interval=record a,b: real; end; var I,J: interval;
---	---	---

Indicați expresia care are valoarea **1** (C++/C), respectiv **true** (Pascal) dacă și numai dacă intersecția intervalelor memorate în variabilele **I** și **J** este mulțimea vidă.

Limbajul C++/C

- a) **(I.a < J.a) && (I.b < J.b) && (I.a < J.b)**
- b) **(I.b <= J.a) || (J.b <= I.a)**
- c) **!(I.b > J.a) && !(J.b > I.a)**
- d) **!(I.b >= J.a) || (J.b < I.a)**
- e) **!(I.b > J.a) || (J.b < I.a)**
- f) **!(I.b > J.a) && (J.b <= I.a)**

Limbajul Pascal

- a) **(I.a < J.a) and (I.b < J.b) and (I.a < J.b)**
- b) **(I.b <= J.a) or (J.b <= I.a)**
- c) **not(I.b > J.a) and not(J.b > I.a)**
- d) **not(I.b >= J.a) or (J.b < I.a)**
- e) **not(I.b > J.a) or (J.b < I.a)**
- f) **not(I.b > J.a) and (J.b <= I.a)**

6. Pentru determinarea în ordine crescătoare a numerelor naturale având exact **2** cifre formate cu elemente din mulțimea **{0, 1, 2}** se utilizează un algoritm backtracking care generează, în ordine, numerele **10, 11, 12, 20, 21, 22**. Dacă se utilizează același algoritm pentru generarea numerelor naturale având exact **3** cifre formate cu elemente din mulțimea **{0, 1, 2}**, precizați câte numere generate sunt pare.

- a) **9** b) **12** c) **18** d) **27** e) **36** f) **40**

7. Subprogramul **f** este definit mai jos.

Limbajul C++/C

```
int f(int n)
{
    if (n==1) return 2;
    else
        return n*(n+1)+f(n-1);
}
```

Limbajul Pascal

```
function f(n : integer) :
integer;
begin
    if (n=1) then f := 2
    else
        f:=n*(n+1)+f(n-1);
end;
```

Precizați ce valoare returnează subprogramul la apelul **f(20)**.

- a) **440** b) **2660** c) **3080** d) **3542** e) **5660** f) **5690**

8. Tabloul unidimensional **A**, cu 5 elemente având valori distincte, memorează cele mai mici 5 numere naturale pătrate perfecte. Tabloul unidimensional **B**, cu 4 elemente având valori distincte, memorează cele mai mici 4 numere naturale prime. Tablourile **A** și **B** sunt sortate descrescător. Se sortează descrescător prin interclasare cele două tablouri **A** și **B** în tabloul unidimensional **C**. Precizați care sunt elementele tabloului **C**, după sortarea prin interclasare a lui **A** și **B**.

- a) (16, 9, 7, 5, 4, 3, 2) b) (16, 9, 7, 5, 4, 3, 2, 1, 0) c) (16, 9, 7, 5, 4, 3)
d) (16, 9, 7, 5, 4, 3, 1) e) (16, 9, 7, 5, 4, 3, 2, 1, 1) f) (16, 9, 7, 5, 4, 2)

9. Variabilele întregi **a**, **b** și **c** memorează inițial valorile 19, 20, respectiv 21. Precizați care sunt valorile lui **a**, **b**, respectiv **c** după apelul **f(a, b, c)** pentru limbajele C++ și Pascal, respectiv **f(a, b, &c)** pentru limbajul C.

Limbajul C++
void f(int a,
int b, int &c)

{
 a= b%c;
 b= a+1;
 c= a%b;
}

Limbajul C
void f(int a,
int b, int *c)

{
 a= b%(*c) ;
 b= a+1;
 ***c= a%b;**
}

Limbajul Pascal
procedure f(a: integer;
b: integer; var c:
integer) ;
begin
 a:= b mod c;
 b:= a+1;
 c:= a mod b;
end;

- a) 19 20 20 b) 19 20 21 c) 19 21 20
d) 20 20 0 e) 20 20 21 f) 20 20 22

10. Se consideră un arbore cu rădăcină având 1026 de noduri etichetate cu numerele naturale de la 1 la 1026. Toate nodurile arborelui respectă relația: **tata[x]=[x/2]** (tatăl nodului **x** este partea întreagă din jumătatea lui **x**). Numărul nodurilor din arbore care au cel mult un descendent direct(fiu) este:

- a) 512 b) 513 c) 514 d) 518 e) 1023 f) 1026

11. Un graf neorientat **G** cu 4 noduri, numerotate de la 1 la 4, are mulțimea muchiilor **{ [1,2] , [2,3] }**. Se construiesc toate subgrafurile distincte ale lui **G** având zero muchii. Două subgrafuri se consideră distincte dacă au mulțimile nodurilor diferite. Precizați câte astfel de subgrafuri distincte ale lui **G** s-au construit (se numără numai subgrafurile lui **G** în care mulțimea muchiilor este mulțimea vidă).

- a) 4 b) 6 c) 9 d) 12 e) 13 f) 16

12. Fie un număr natural nenul n . Dorim să numărăm în câte cifre consecutive de zero se termină produsul $1 \cdot 2 \cdot 3 \cdot \dots \cdot n$. Dacă trebuie să calculăm acest număr de zerouri consecutive cel mai eficient din punct de vedere al timpului de execuție, alegem să utilizăm un algoritm bazat pe cea mai restrictivă variantă, având complexitatea timp:
- a) $O(1)$, algoritm bazat pe o formulă matematică b) $O(\log n)$, algoritm logaritmic c) $O(n)$, algoritm liniar
d) $O(n^2)$, algoritm pătratic e) $O(n^3)$, algoritm cubic f) $O(2^n)$, algoritm exponențial
13. Se consideră graful orientat cu 4 vârfuri, etichetate cu numere de la 1 la 4, având mulțimea arcelor $\{(1, 2), (3, 2), (4, 1), (4, 2), (4, 3)\}$. Indicați numărul minim de arce care trebuie adăugate în acest graf orientat astfel încât noul graf să devină tare conex.
- a) 5 b) 4 c) 3 d) 2 e) 1 f) 0
14. Precizați câte grafuri neorientate distincte, cu nodurile etichetate de la 1 la 8, se pot construi, știind că în fiecare graf construit se respectă simultan proprietățile de mai jos:
- 1) Fiecare nod etichetat cu un număr prim este adiacent cu nodul 8.
2) Nu există nicio muchie $[x, y]$ cu ambele extremități x și y numere impare.
- Două grafuri neorientate se consideră distincte dacă au matricele de adiacență diferite.
- a) 2^9 b) 2^{17} c) 4^9 d) 2^{28} e) 4^{17} f) 2^{56}
15. Tabloul unidimensional $\mathbf{v}$ are 33 de elemente, numerotate de la 1 la 33. Valorile elementelor din $\mathbf{v}$ sunt numere naturale. Tabloul $\mathbf{v}$ conține, începând cu indicele 1, primii 33 de termeni ai șirului de numere naturale: $(0, 1, 4, 9, 61, 52, 63, 94, 46, 18, 1, \dots)$. Deduceți regula de generalizare după care s-au construit termenii șirului și precizați câte elemente ale lui $\mathbf{v}$ se termină cu cifra 1.
- a) 26 b) 17 c) 12 d) 9 e) 8 f) 4

Varianta 32

1. Variabila **a** memorează un număr natural care **nu** este multiplu de 3. Expresia care are totdeauna valoarea egală cu o treime din **a** este:

Limbajul C++/C	a) $a / (3 * 2) / 2$ c) $a / 2 / 3 + a / 3 / 2$ e) $a / 3 * a / 2$	b) $a / 3 + a / 2$ d) $a / (2 / 3) / 3$ f) $a / 2 / 3 * 2$
Limbajul Pascal	a) $a \text{ div } (3 * 2) \text{ div } 2$ c) $a \text{ div } 2 \text{ div } 3 + a \text{ div } 3 \text{ div } 2$ e) $a \text{ div } 3 * a \text{ div } 2$	b) $a \text{ div } 3 + a \text{ div } 2$ d) $a \text{ div } (2 \text{ div } 3) \text{ div } 3$ f) $a \text{ div } 2 \text{ div } 3 * 2$

2. Variabilele **a** și **b** memorează numere naturale nenule. Se consideră următoarea secvență de program:

Limbajul C++/C	Limbajul Pascal
<pre>for(i:=a*b; i>=1; i--) if(i%a==0 && i%b==0) c=i; cout<<c; printf("%d",c);</pre>	<pre>for i:=a*b downto 1 do if i mod a=0 and i mod b=0 then c:=i; write(c);</pre>

În urma executării secvenței de program alăturate, variabila **c** are valoarea:

- a) cel mai mic multiplu comun al numerelor **a** și **b**;
b) cel mai mare număr mai mic decât produsul numerelor **a** și **b**, care divide pe **a** și pe **b**;
c) cel mai mic număr mai mare decât produsul numerelor **a** și **b**, care este divizibil cu **a** și cu **b**;
d) cel mai mare divizor comun al numerelor **a** și **b**;
e) suma divizorilor numerelor **a** și **b**;
f) produsul divizorilor numerelor **a** și **b**.

3. Variabila **x** reține un număr natural mai mic decât 19, iar **i** și **y** sunt variabile de tip întreg. Se consideră următoarea secvență de program:

Limbajul C++/C	Limbajul Pascal
<pre>for(i=1; i<=9; i++) if((x-i)>=0 && (x-i)<=9) { y=10*i+(x-i); cout<<y<<' '; printf("%d ",y); }</pre>	<pre>for i:=1 to 9 do if (x-i)>=0 and (x-i)<=9 then begin y:=10*i+(x-i); write(y, ' '); end;</pre>

În urma executării secvenței de program alăturate, se afișează:

- a) numerele naturale de două cifre care au suma cifrelor egală cu **x**;
b) numerele naturale care au suma cifrelor egală cu **x**;
c) numere naturale mai mari decât 10 și mai mici decât **x**;
d) numere naturale cu cifre distincte, mai mici decât **x**;
e) numere naturale cu cifre distincte, mai mari decât **x**;

f) numerele naturale de cel puțin două cifre care au suma cifrelor egală cu **x**.

4. Variabila **x** memorează notele obținute de un elev la cele trei probe de Bacalaureat, note cu două zecimale. Declarația corectă a variabilei **x** este:

Limbajul C++/C	a) <code>char x[2];</code> c) <code>float x;</code> e) <code>int x[2];</code>	b) <code>int x;</code> d) <code>float x[3];</code> f) <code>float x[2][3];</code>
Limbajul Pascal	a) <code>var x:string[2];</code> c) <code>var x:real;</code> e) <code>var x:array[1..2] of byte;</code>	b) <code>var x:byte;</code> d) <code>var x:array[1..3] of real;</code> f) <code>var x:array[1..2,1..3] of real;</code>

5. Într-un graf neorientat, cu 10 noduri, fiecare nod are gradul 2. Numărul maxim de componente conexe din care poate fi format graful este:

a) 1 b) 2 c) 3 d) 4 e) 5 f) 6

6. Pentru a calcula cel mai mare divizor comun pentru numerele naturale nenule **a** și **b**, se consideră următoarea secvență de program:

Limbajul C++/C	Limbajul Pascal
<code>while (a!=b) if (a>b) a=a-b; else b=b-a;</code>	<code>while a<>b do if a>b then a:=a-b else b:=b-a;</code>

Algoritmul este:

a) eficient b) ineficient c) incorect sintactic d) incorect semantic e) greșit f) infinit

7. Se consideră graful neorientat **G** cu 5 noduri, reprezentat prin matricea de adiacență alăturată.

```

0  1  0  0  1
1  0  1  1  1
0  1  0  1  0
0  1  1  0  0
1  1  0  0  0

```

Afirmația adevărată este:

- a) Graful **G** conține două componente conexe;
b) Orice subgraf a lui **G** format din 3 noduri este arbore;
c) Graful **G** este hamiltonian;
d) Graful **G** este eulerian;
e) Graful **G** este arbore;
f) Graful **G** nu este eulerian.

8. Variabilele **n** și **i** memorează numere naturale întregi. În următoarea secvență de program, **v** este un tablou unidimensional cu **n** elemente:

Limbajul C++/C	Limbajul Pascal
<code>i=0; while (i<n) v[i++]=i*i*i;</code>	<code>i:=1; while i<=n do begin v[i]:=i*i*i;inc(i) end;</code>

Numărul de repetări ale secvenței de instrucțiuni din **while** este:

a) $n+1$ b) $n-1$ c) n d) 0 e) 1 f) $3 \cdot n$

9. Un elev folosește metoda backtracking pentru a genera submulțimile mulțimii $\{1, 2, 5, 6, 9\}$. Numărul de submulțimi generate, care obligatoriu conțin elementul 2 și **nu** conțin elementul 6, este?

a) 16 b) 8 c) 7 d) 6 e) 4 f) 2

10. Subprogramul **f**, cu parametrii **a** și **b** numere întregi ($a < b$), returnează numărul de numere pare din intervalul $[a, b]$. Expresia care are valoarea 1 (C++/C) / **True** (Pascal), pentru orice numere **a** și **b** care **nu** au aceeași paritate este:

Limbajul	a) $f(a, b) == f(a, b+1)$	b) $f(a, b) == (b-a) / 2$
C++/C	c) $f(a, b) == (b-a+1) / 2$	d) $f(a, b) == b-a$
	e) $f(a, b) == b-a+1$	f) $f(a, b) == (b-a-1) / 2$
Limbajul	a) $f(a, b) = f(a, b+1)$	b) $f(a, b) = (b-a) \text{ div } 2$
Pascal	c) $f(a, b) = (b-a+1) \text{ div } 2$	d) $f(a, b) = b-a$
	e) $f(a, b) = b-a+1$	f) $f(a, b) = (b-a-1) \text{ div } 2$

11. Se consideră un arbore, care are rădăcina pe nivelul 1 și orice nod de pe nivelul i are exact $i+1$ descendenți direcți, cu excepția nodurilor de pe nivelul 4 care sunt noduri terminale. Numărul de frunze ale arborelui este:

a) 120 b) 30 c) 24 d) 8 e) 6 f) 4

12. Se consideră următorul subprogram:

Limbajul C++	Limbajul C	Limbajul Pascal
void f(int x, int *y)	void f(int x, int &y)	procedure f(x:integer; var
{	{	y:integer);
y=x+y;	*y=x+y;	begin
x=x+y;	x=x+y;	y:=x+y;
}	}	x:=x+y
		end;

Dacă valoarea variabilei **a** înainte de apel este 2, care este valoarea sa după apelul:

Limbajul C++: f(a, a)	Limbajul C: f(a, &a)	Limbajul Pascal: f(a, a)
a) 12 b) 10 c) 8 d) 6 e) 4 f) 2		

13. Subprogramul **f**, cu doi parametri întregi **x** și **y**, returnează valoarea celui mai mare divizor comun al numerelor **x** și **y**. Expresia prin care se calculează cel mai mare divizor comun al numerelor **x**, **y** și **z** este:

a) $f(x, y) + f(y, z)$ b) $f(x, y, z)$ c) $f(x, y) * z$
d) $f(x, y) * f(y, z)$ e) $f(x * y, z)$ f) $f(f(x, y), z)$

14. Pentru variabilele întregi **x** și **y**, subprogramul **mic(x, y)** întoarce cel mai mic număr dintre **x** și **y**, subprogramul **mare(x, y)** întoarce cel mai mare număr dintre **x** și **y**, iar subprogramul **p(x, y)** întoarce valoarea puterii lui **x** cu exponent **y**. Pentru ca **u** și **v** să fie cel mai mare divizor comun, respectiv cel mai mic multiplu comun al numerelor 6^x și 6^y , atunci subprogramele **f1**, **f2**, **f3** și **f4** din instrucțiunile:

Limbajul C++/C	Limbajul Pascal
-----------------------	------------------------

$u = p(2, f1(x, y)) * p(3, f2(x, y));$ $v = p(2, f3(x, y)) * p(3, f4(x, y));$	$u := p(2, f1(x, y)) * p(3, f2(x, y));$ $v := p(2, f3(x, y)) * p(3, f4(x, y));$
--	--

sunt, respectiv:

- | | |
|---|---|
| <p>a) mic, mic, mare, mare;</p> <p>c) mare, mare, mic, mic;</p> <p>e) mare, mic, mic, mare;</p> | <p>b) mic, mare, mic, mare;</p> <p>d) mare, mic, mare, mic;</p> <p>f) mic, mare, mare, mic.</p> |
|---|---|

15. Numărul de cicluri hamiltoniene distincte într-un graf neorientat complet cu n noduri ($n \geq 3$) este (două cicluri se consideră distincte dacă diferă prin cel puțin o muchie):

- a) $\frac{n(n-1)}{2}$ b) $\frac{(n-1)!}{2}$ c) $\frac{(n-2)(n-1)}{2}$ d) $n - 2$ e) $\frac{(n+1)!}{2}$ f) $\frac{(n+2)(n+1)}{2}$

Varianta 33

1. Variabilele **x** și **y** memorează numere reale. Se consideră următoarea secvență de program:

Limbajul C++/C	Limbajul Pascal
<code>y=x;x=x*x;</code>	<code>y:=x;x:=x*x;</code>
<code>if (x<y) cout<<"DA";</code>	<code>if x<y then write('DA');</code>
<code>lprintf("DA");</code>	

Executarea secvenței de program alăturate afișează **DA** pentru valori inițiale ale lui **x**:

- | | |
|--|--|
| <p>a) strict pozitive subunitare;</p> <p>c) strict negative subunitare;</p> <p>e) strict pozitive;</p> | <p>b) strict pozitive supraunitare;</p> <p>d) strict negative supraunitare;</p> <p>f) strict negative.</p> |
|--|--|
2. Variabilele **n** și **k** memorează numere naturale nenule. Expresia prin care se poate calcula cel mai mare număr natural divizibil cu **k**, număr care să fie mai mic sau egal cu **n** este:

	Limbajul a) Nu există formulă.	b) $(k*n) / k$
C++/C	c) $n\%k + n/k$	d) $(k+n) / k$
	e) $n - n/k$	f) $n - n\%k$
Limbajul	a) Nu există formulă.	b) $(k * n) \text{ div } k$
Pascal	c) $n \bmod k + n \text{ div } k$	d) $(k + n) \text{ div } k$
	e) $n - n \text{ div } k$	f) $n - n \bmod k$

3. Variabilele **n** și **i** memorează numere naturale nenule. Se consideră următoarea secvență de program:

Limbajul C++	Limbajul C	Limbajul Pascal
<code>for (i=1;i<=5;i++)</code>	<code>for (i=1;i<=5;i++)</code>	<code>for i:=1 to 5 do</code>
<code>n=n*i;</code>	<code>n=n*i;</code>	<code>n:=n*i;</code>
<code>cout<<n;</code>	<code>printf("%d",n);</code>	<code>write(y);</code>

Valoarea inițială a variabilei **n** pentru care executarea secvenței de program alăturate afișează **360** este:

- | | | | | | |
|------|------|------|------|------|------|
| a) 0 | b) 1 | c) 2 | d) 3 | e) 4 | f) 5 |
|------|------|------|------|------|------|
4. Variabilele **a** și **b** memorează numere naturale nenule. Se consideră următoarea secvență de program:

Limbajul C++/C	Limbajul Pascal
<code>a=0;b=...;</code>	<code>a:=0;b:=...;</code>
<code>while (b<10)</code>	<code>while b<10 do</code>
<code>{</code>	<code>begin</code>
<code> a=a+b; b++;</code>	<code> a:=a+b;b:=b+1</code>
<code>}</code>	<code>end;</code>
<code>cout<<a; printf("%d",a);</code>	<code>write(a);</code>

Valoarea care poate înlocui punctele de suspensie din secvența alăturată, astfel încât valoarea afișată să fie **35** este:

- | | | | | | |
|------|------|------|------|------|------|
| a) 2 | b) 3 | c) 4 | d) 5 | e) 6 | f) 7 |
|------|------|------|------|------|------|
5. Un graf neorientat are gradele vârfurilor: 2, 3, 3, 2, 4. Numărul de muchii ale grafului este:

- a) 5 b) 6 c) 7 d) 8 e) 9 f) 10

6. Variabilele i și j memorează numere naturale nenule. Se consideră următoarea secvență de program:

Limbajul C++/C	Limbajul Pascal
<code>for(i=0; i<=9; i++)</code>	<code>for i:=0 to 9 do</code>
<code> for(j=0; j<=9; j++)</code>	<code> j:=0 to 9 do</code>
<code> a[i][j]=(2*i+3*j)%10;</code>	<code> a[i][j]:=(2*i+3*j)%10;</code>

Suma elementelor de pe diagonala principală a tabloului bidimensional constuit este:

- a) 10 b) 25 c) 50 d) 45 e) 46 f) 100

7. Numărul **maxim** de comparații pentru ordonarea descrescătoare a valorilor celor 100 de componente ale tabloului unidimensional v , ordonare realizată prin metoda bulelor, este:

- a) 100 b) 4950 c) 9701 d) 9900 e) 9999 f) 10000

8. Subprogramul $f(a, b)$ returnează media aritmetică a numerelor reale a și b . Pentru a , b , c și d numere reale, instrucțiunea care atribuie variabilei a suma dintre media aritmetică a numerelor b și c și media aritmetică a numerelor c și d este:

Limbajul C++/C	a) $a = f(b, f(c, d));$	b) $a = f(f(b, c), d);$
	c) $a = (b + c + d) / 2;$	d) $a = f(b, c) + f(b, d);$
	e) $a = f(b, d) + c;$	f) $a = (f(b, c) + f(b, d)) / 2;$
Limbajul Pascal	a) $a := f(b, f(c, d));$	b) $a := f(f(b, c), d);$
	c) $a := (b + c + d) / 2;$	d) $a := f(b, c) + f(b, d);$
	e) $a := f(b, d) + c;$	f) $a := (f(b, c) + f(b, d)) / 2;$

9. Se consideră un graf neorientat G , cu 5 noduri, reprezentat prin matricea de adiacență alăturată.

0	1	0	0	1
1	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	1	0	0	0

Afirmația adevărată este:

- a) G este graf hamiltonian și graf eulerian;
b) G este graf hamiltonian, dar **nu** este graf eulerian;
c) G **nu** este graf hamiltonian, dar este graf eulerian;
d) G **nu** este graf hamiltonian, **nici** graf eulerian;
e) G este graf hamiltonian;
f) Toate afirmațiile de mai sus sunt **false**.

10. Pentru a determina toate modalitățile de a scrie pe 9 ca sumă de numere naturale nenule distincte (abstracție făcând de ordinea termenilor), un elev folosește metoda backtracking generând, în această ordine, toate soluțiile: $1+2+6$, $1+3+5$, $1+8$, $2+3+4$, $2+7$, $3+6$ și $4+5$. Aplicând exact aceeași metodă, el determină soluțiile pentru scrierea lui 12. Numărul soluții de forma $3 + \dots$ este:

- a) 0 b) 1 c) 2 d) 4 e) 6 f) 7

11. Fie G un arbore cu n ($n > 1$) noduri și $d_1 \geq d_2 \geq d_3 \geq \dots \geq d_n \geq 1$ gradele nodurilor sale. Afirmația adevărată este:

$$\begin{array}{lll} \text{a) } \sum_{i=1}^n d_i = 2n - 2 & \text{b) } \sum_{i=1}^n d_i = 2n - 1 & \text{c) } \sum_{i=1}^n d_i = 2n \\ \text{d) } \sum_{i=1}^n d_i = 2n + 1 & \text{e) } \sum_{i=1}^n d_i = 2n + 2 & \text{f) } \sum_{i=1}^n d_i = n \end{array}$$

12. Funcția f primește două valori întregi prin intermediul a doi parametri și returnează suma tuturor cifrelor celor două numere. De exemplu, $f(173, 608)$ returnează 25. Apelul funcției f care determină suma cifrelor unui număr întreg n este:

$$\text{a) } f(1, 1) \quad \text{b) } f(n, 0) \quad \text{c) } f(n, 1) \quad \text{d) } f(n, n) \quad \text{e) } f(1, n) \quad \text{f) } f(n, n-1)$$

13. Într-o coadă, inițial vidă, la fiecare pas k se introduc $3k$ valori și se extrag $k+2$ valori. După executarea primilor 9 pași, în coadă se află un număr de elemente egal cu:

$$\text{a) } 9 \quad \text{b) } 36 \quad \text{c) } 72 \quad \text{d) } 75 \quad \text{e) } 79 \quad \text{f) } 172$$

14. Se consideră un graf neorientat conex cu n noduri și m muchii. Pentru a obține exact 2 componente conexe, numărul minim de muchii care trebuie eliminate este egal cu:

$$\begin{array}{lll} \text{a) gradul minim din graf} & \text{b) gradul maxim din graf} & \text{c) } m-1 \\ \text{d) } n-1 & \text{e) } \frac{m(m-1)}{2} & \text{f) } \frac{n(n-1)}{2} - m \end{array}$$

15. Numărul de elemente nule ale matricei de adiacență asociată unui arbore cu n noduri este:

$$\text{a) } n^2 \quad \text{b) } n^2+1 \quad \text{c) } n(n-1)+n \quad \text{d) } n^2-n-2 \quad \text{e) } n(n-1)-n \quad \text{f) } n^2-2n+2$$

- e) u și v sunt egale numai dacă $y = 1$ f) u și v sunt egale numai dacă $x+y+z=3$
5. Numărul minim de noduri dintr-un graf neorientat cu 12 muchii, fără noduri izolate, graf format din exact 3 componente conexe este:
a) 7 b) 8 c) 9 d) 10 e) 11 f) 12
6. Se consideră tabloul unidimensional v cu n elemente (n număr natural, $n \geq 2$). Subprogramul $f(v, i, j)$ inversează ordinea elementelor aflate pe pozițiile $i, i+1, \dots, j-1, j$ ($1 \leq i < j \leq n$). Secvența de program care inversează, în v , doar $v[i]$ și $v[j]$ este:
a) $f(v, i, j); f(v, i+1, j-1);$ b) $f(v, i, j); f(v, i-1, j+1);$
c) $f(v, i, j); f(v, i+1, j+1);$ d) $f(v, i+1, j-1); f(v, i, j);$
e) $f(v, i+1, j+1); f(v, i, j);$ f) $f(v, i, j); f(v, i+1, j-1);$
7. Se consideră un arbore. Referitor la un lanț elementar care unește două noduri distincte a și b , afirmația adevărată este:
a) Este unic, dacă și numai dacă a sau b este frunză. b) Sigur conține rădăcina arborelui.
c) Este unic, oricare ar fi a și b . d) Nu poate trece prin rădăcina arborelui.
e) Este unic, dacă și numai dacă a sau b este rădăcină. f) Nu este unic oricare ar fi a și b .
8. Variabilele a, b, c memorează numere naturale nenule. Instrucțiunea
Limbajul C++/C: $c = b - b \% a$ | **Limbajul Pascal:** $c := b - b \bmod a$
atribuie variabilei c o valoare care reprezintă:
a) cel mai mic număr natural mai mare sau egal cu a și care este divizibil cu b ;
b) cel mai mic număr natural mai mare sau egal cu b și care este divizibil cu a ;
c) cel mai mic număr natural mai mare sau egal cu b și care este nedivizibil cu a ;
d) cel mai mare număr natural mai mic sau egal cu b și care este divizibil cu a ;
e) cel mai mare număr natural mai mic sau egal cu a și care este divizibil cu b ;
f) cel mai mare număr natural mai mic sau egal cu a și care este nedivizibil cu b .
9. Se consideră subprogramul $f1$:
Limbajul C++/C

```
void f1(int a[50][50], int n, int m)
{
    int i, j;
    for (i=1; i<=n-1; i++)
        for (j=i+1; j<=n; j++)
            if (a[i][2]>a[j][2]) f2(a, n, m, i, j);
}
```

Limbajul Pascal

```
procedure f1(var a:array[1..5, 1..50] of integer; n, m:integer);
var i, j:integer;
begin
    for i:=1 to n-1 do
        for j:=i+1 to n do
            if a[i][2]>a[j][2] then f2(a, n, m, i, j);
end;
```

Subprogramul **f2** realizează interschimbarea elementelor liniilor **i** și **j** ale tabloului transmis prin parametrul **a**, care are **n** linii și **m** coloane. Numerotarea liniilor și a coloanelor începe de la **1**.

Pentru a ordona crescător numerele de pe a doua coloană a tabloului **a**, numărul de apeluri ale subprogramului **f2** necesar este:

- a) cel puțin $\frac{n(n-1)}{2}$ b) cel mult $\frac{m(m-1)}{2}$ c) exact $\frac{n(n-1)}{2}$
d) cel mult $\frac{n(n-1)}{2}$ e) cel puțin $\frac{m(m-1)}{2}$ f) exact $\frac{m(m-1)}{2}$

10. Generarea tuturor tablourilor bidimensionale de ordin **n**, cu elemente **6** și **9**, cu proprietatea că pe fiecare linie și pe fiecare coloană există un singur element egal cu **9**, se poate realiza utilizând metoda backtracking. Algoritmul utilizat este echivalent cu algoritmul de generare a:

- a) aranjamentelor b) combinațiilor c) permutărilor
d) produsului cartezian e) submulțimilor f) problemei celor **n** dame

11. Se consideră următorul subprogram:

Limbaajul C++	Limbaajul C	Limbaajul Pascal
<pre>void f(char a, char &b) {char x=a;a=b;b=x;}</pre>	<pre>void f(char a, char *b) {char x=a;a=*b;*b=x;}</pre>	<pre>procedure f(a:char;var b:char); var x:char; begin x:=a;a:=b;b:=x end;</pre>

Dacă, înainte de apel, **a='a'** și **b='b'**, după executarea secvenței de program alăturate se afișează:

Limbaajul C++	Limbaajul C	Limbaajul Pascal
<pre>f(a,b); cout<<a<<' '<<b;</pre>	<pre>f(a,&b); printf("%c %c",a,b)</pre>	<pre>f(a,b); write(a,' ',b);</pre>
a) a a b) b b	c) b a d) a b	e) aa f) bb

12. Numim **graf complementar** al unui graf neorintat **G1** graful neorientat **G2** cu aceeași mulțime a nodurilor ca și **G1** și cu proprietatea că două noduri sunt adiacente în **G2** dacă și numai dacă **nu** sunt adiacente în **G1**. Dacă **G1** are **n** noduri și **m** muchii, numărul de muchii pentru **G2** este:

- a) minim $\frac{n(n-1)}{2} - m$ b) exact $\frac{n(n-1)}{2} - m$ c) maxim $\frac{n(n-1)}{2} - m$
d) minim $n-m$ e) exact $n-m$ f) maxim $n-m$

13. Subprogramul **f(a,b)** returnează cel mai mare divizor prim al numărului natural **a**, divizor mai mic sau egal cu **b** ($a \geq 3$, $2 \leq b \leq a$). Expresia care are valoarea **1** (C++/C) / **True** (Pascal), dacă și numai dacă **a** este un număr **prim** este:

Limbaajul	a) $f(a, a-1) == 2$	b) $f(a, a) == 2$	c) $f(a, a) == a$
C++/C	d) $f(a, a/2) == a/2$	e) $f(a, a) == 1$	f) $f(a, a) == a/2$
Limbaajul	a) $f(a, a-1) = 2$	b) $f(a, a) = 2$	c) $f(a, a) = a$
Pascal:	d) $f(a, a \text{ div } 2) = a \text{ div } 2$	e) $f(a, a) = 1$	f) $f(a, a) = a \text{ div } 2$

14. Fie G un graf orientat, cu n noduri și m arce. Dacă s_1 și s_2 reprezintă suma gradelor interioare, respectiv exterioare ale grafului G , afirmația **falsă** este:
- a) $s_1 = s_2$ b) $s_1 + s_2 = 2m$ c) dacă G este graf complet, atunci $s_1 + s_2 = n(n-1)$
d) $s_1 \leq s_2$ e) $s_1 \geq s_2$ f) dacă G este graf complet, atunci $s_1 + s_2 = m(m-1)$
15. Un arbore binar este un arbore cu rădăcină în care fiecare nod are cel mult 2 descendenți direcți (fii). Un arbore binar complet cu n noduri are un număr de niveluri egal cu:
- a) $\lceil \log_2 n \rceil - 1$ b) $\lceil \log_2 n \rceil$ c) $\lceil \log_2 n \rceil + 1$
d) $\lceil \log_2 (n-1) \rceil$ e) $\lceil \log_2 (n+1) \rceil$ f) $\lceil \log_2 n \rceil - 1$

Varianta 35

1. Variabilele **n** și **c** memorează numere naturale nenule. Instrucțiunea care inserează cifra **c** în fața ultimei cifre a lui **n** este:

Limbajul C++/C	a) $n = (n \% 10 * 10 + c) * 10 + n / 10;$ c) $n = (n / 10 + c) * 10 + n \% 10;$ e) $n = n / 10 * 10 + c * 10 + n \% 10;$	b) $n = (n / 10 * 10 + c) * 10 + n \% 10;$ d) $n = n / 10 + c + n \% 10;$ f) $n = (n \% 10 + c) * 10 + n / 10;$
Limbajul Pascal	a) $n := (n \bmod 10 * 10 + c) * 10 + n \operatorname{div} 10;$ c) $n := (n \operatorname{div} 10 + c) * 10 + n \bmod 10;$ e) $n := n \operatorname{div} 10 * 10 + c * 10 + n \bmod 10;$	b) $n := (n \operatorname{div} 10 * 10 + c) * 10 + n \bmod 10;$ d) $n := n \operatorname{div} 10 + c + n \bmod 10;$ f) $n := (n \bmod 10 + c) * 10 + n \operatorname{div} 10;$

2. Variabilele **a** și **b** memorează numere naturale. Se consideră următoarea secvență de program:

Limbajul C++/C	Limbajul Pascal
<pre> b=2; for(a=5; a<=10; a++) { a=a+b; b=a+b; } cout<<a+b; printf("%d",a+b); </pre>	<pre> b:=2; for a:= 5 to 10 do begin a:=a+b; b:=a+b end; write(a+b); </pre>
În urma executării secvenței de program alăturate se afișează:	
a) 18 b) 26 c) 28 d) 44 e) 48 f) 52	

3. Variabilele **i**, **j** și **k** memorează numere naturale. Se consideră următoarea secvență de program:

Limbajul C++/C	Limbajul Pascal
<pre> for(i=1; i<=10; i++) { for(j=1; j<=i; j++) cout<<j; printf("%d",j); for(k=9; k>0; k--) cout<<k; printf("%d",k); } </pre>	<pre> for i:=1 to 10 do begin for j:=1 to i do write(j); for k:=9 downto 1 do write(k) end; end; </pre>
Numărul de execuții ale instrucțiunii care afișează valoarea variabilei k este:	
a) 495 b) 90 c) 60 d) 55 e) 10 f) 9	

4. Pentru un graf neorientat cu 9 muchii și 12 noduri, numărul minim de componente conexe este:

a) 1 **b) 2** **c) 3** **d) 4** **e) 5** **f) 6**

5. Algoritmul lui Euclid este utilizat pentru:

a) calculul numărului de multipli ai unui număr natural;
b) descompunerea în factori primi a unui număr natural;
c) calculul celui mai mare divizor comun a două numere naturale;

- d) calculul numărului de divizori ai unui număr natural;
- e) suma divizorilor unui număr natural;
- f) suma divizorilor proprii ai unui număr natural.

6. Variabilele **x, y, z, s** și **p** memorează numere reale. Se consideră următoarea secvență de program:

Limbaajul C++/C	Limbaajul Pascal
<code>if(x>y) if(y>z) if(z>x)</code>	<code>if x>y then if y>z then if z>x then</code>
<code>s=x+y+z; else p=x*y*z;</code>	<code>s:=x+y+z else p:=x*y*z;</code>

Secvența de program echivalentă cu ea, care să conțină o singură instrucțiune de decizie, este:

Limbaajul C++/C a) <code>if(x>y y>z) s=x+y+z;</code> b) <code>if(x>y && y>z) s=x+y+z;</code> c) <code>if(x>y && y>z) s=x+y+z;</code> d) <code>if(x>y && y>z) p=x*y*z;</code> e) <code>if(x>y && y>z) s=x+y+z;</code> f) <code>if(x>y y>z) p=x*y*z;</code> <code>else p=p*y*z;</code>	Limbaajul Pascal a) <code>if (x>y) or (y>z) then s:=x+y+z;</code> b) <code>if (x>y) and (y>z) then s:=x+y+z;</code> c) <code>if (x>y) and (y>z) then s:=x+y+z;</code> d) <code>if (x>y) and (y>z) then p:=x*y*z;</code> e) <code>if (x>y) and (y>z) then s:=x+y+z</code> <code>else p:=p*y*z;</code> f) <code>if (x>y) or (y>z) then p:=x*y*z;</code>
--	---

7. Numărul de interschimbări care se efectuează în cazul sortării descrescătoare a șirului de numere consecutive 0, 1, 2, 3, ..., 8, 9, 10 prin metoda bulelor este:
- a) 0 b) 10 c) 11 d) 45 e) 55 f) 121

8. Fie **a** un tablou bidimensional cu 45 linii (numerotate de la 1 la 45) și 45 coloane (numerotate de la 1 la 45). Expresia care calculează numărul de ordine al elementului de pe linia **i** și coloana **j** (a câta valoare este acesta, pornind din colțul din stânga sus, de la prima spre ultima linie, pe fiecare linie elementele numărându-se de la stânga la dreapta) este:

a) $i*45+j-1$	b) $(i-1)*45+j$	c) $(j-1)*45+i$
d) $j*45+i-1$	e) $(i+1)*45+j$	f) $(j+1)*45+i$

9. Se consideră algoritmul care determină toate permutările distincte de **n** obiecte (numerotate de la 1 la **n**), în care pe orice poziție de rang par se află o valoare pară. De exemplu, pentru **n=5**, primele trei permutări generate în ordine lexicografică sunt: (1, 2, 3, 4, 5), (1, 2, 5, 4, 3), (1, 4, 3, 2, 5).

Pentru **n=4**, numărul total de astfel de permutări este:

a) 12 b) 10 c) 8 d) 7 e) 6 f) 4

10. Subprogramul **f** primește prin parametrii **a** și **b** două valori întregi ($a \leq b$) și returnează numărul de numere prime din intervalul închis $[a, b]$. Expresia care are valoarea 1 (**C++/C**) / **True (Pascal)**, numai dacă valoarea întreagă **x** ($x > 5$) este număr prim este:

a) `f(x-1, x) == f(x, x+1)` b) `f(x, x) == 1`

Limbaajul C++/C	c) $f(2, x) \neq f(2, x-1)$ e) $f(2, x) == f(2, x-1)$	d) $f(2, x) \neq f(2, x+1)$ f) $f(2, x) == f(2, x+1)$
Limbaajul Pascal	a) $f(x-1, x) = f(x, x+1)$ c) $f(2, x) <> f(2, x-1)$ e) $f(2, x) = f(2, x-1)$	b) $f(x, x) = 1$ d) $f(2, x) <> f(2, x+1)$ f) $f(2, x) = f(2, x+1)$

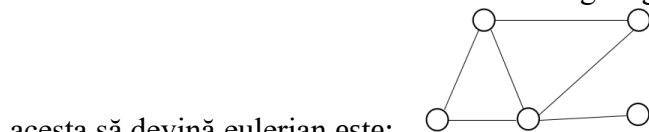
11. Se consideră următoarea secvență de program:

Limbaajul C++	Limbaajul C	Limbaajul Pascal
<pre>int a,b; void f(int x,int &y) {int b=x;y+=b;x=y;} int main() { a=20;b=23; f(a,b); cout<<a<<' '<<b; return 0; }</pre>	<pre>int a,b; void f(int x,int *y) {int b=x;*y=*y+b;x=*y;} int main() { a=20;b=23; f(a,&b); printf("%d %d",a,b); return 0; }</pre>	<pre>var a,b:integer; procedure f(x:integer; var y:integer); var b:integer; begin b:=x;y:=y+b;x:=y end; begin a:=20;b:=23; f(a,b);write(a,' ',b) end.</pre>

În urma executării secvenței de program alăturate se afișează:

- a) 43 43 b) 23 23 c) 20 23 d) 23 43 e) 20 43 f) 23 20

12. Numărul minim de muchii care trebuie adăugate grafului din figura alăturată, astfel încât



acesta să devină eulerian este:

- a) 0 b) 1 c) 2 d) 3 e) 4 f) 5

13. Subprogramul **minim** returnează cifra minimă a unui număr natural. Pentru o variabilă **x**, ce memorează o valoare naturală de cel mult 2 cifre, subprogramul este apelat într-o secvență de forma **Limbaajul C++/C**: `if (minim(x)+minim(x*x*x))==0) nr++;`

Limbaajul Pascal: `if minim(x)+minim(x*x*x)=0 then nr:=nr+1;`

Varianta pentru un antet corect al subprogramului este:

Limbaajul C++/C	Limbaajul Pascal
a) <code>int minim(long u)</code>	a) <code>function minim(u:longint):integer;</code>
b) <code>int minim(long x*x*x)</code>	b) <code>function minim(x*x*x:longint):integer;</code>
c) <code>int minim(int x, int y)</code>	c) <code>function minim(x,y:integer):integer;</code>
d) <code>void minim(long u)</code>	d) <code>procedure minim(u:longint);</code>
e) <code>void minim(int x, int y)</code>	e) <code>procedure minim(x,y:longint);</code>
f) <code>void minim(long x*x*x)</code>	f) <code>procedure minim(x*x*x:longint);</code>

14. Un arbore binar este un arbore cu rădăcină în care fiecare nod are cel mult 2 descendenți direcți (fii).

Un arbore binar complet, cu **h** niveluri, are un număr de noduri egal cu:

- a) $2h$ b) $2h+1$ c) $2h-1$ d) 2^h-1 e) 2^h f) 2^{h+1}

15. Fie G un graf neorientat, cu n noduri și p componente conexe.

Numărul maxim de muchii este:

a) $\frac{n(n-1)}{2}$ b) $\frac{(n-p)(n-p+1)}{2}$ c) $\frac{n(n+1)}{2}$ d) $\frac{(n+p)(n+p+1)}{2}$ e) $\frac{p(p-1)}{2}$ f) $\frac{p(p+1)}{2}$

Varianta 36

1. Variabilele **x** și **y** memorează numere întregi. Se consideră următoarea secvență de program:

Limbajul C++/C

```
x=2020/7;
y=123%10*3/8;
cout<<x<<' '<<y
|printf("%d
%d",x,y);
```

Limbajul Pascal

```
x:=2020 div 7;
y:=123 mod 10*3 div 8;
write(x,' ',y);
```

După executarea secvenței de program alăturate, variabilele **x** și **y** au valorile:

- a) 66 0 b) 66 1 c) 202 0 d) 202 1 e) 288 0 f) 288 1

2. Se consideră următoarea expresie:

Limbajul C++/C: $(x==y) == (y==z)$ **Limbajul Pascal:** $(x=y) = (y=z)$

Expresia dată are valoarea **0(C++/C)/False(Pascal)** dacă și numai dacă cele trei variabile întregi **x**, **y** și **z** sunt:

- a) toate trei egale

Limbajul C++/C

c) $(x==y \ \&\& \ y!=z) \ || \ (x!=y \ \&\& \ y==z)$

d) $(x==y \ \&\& \ y!=z) \ \&\& \ (x!=y \ \&\& \ y==z)$

e) $(x==y \ || \ y!=z) \ || \ (x!=y \ || \ y==z)$

f) $(x==y \ || \ y!=z) \ \&\& \ (x!=y \ || \ y==z)$

- b) neinițializate

Limbajul Pascal

c) $(x=y \ \text{and} \ y<>z) \ \text{or} \ (x<>y \ \text{and} \ y=z)$

d) $(x=y \ \text{and} \ y<>z) \ \text{and} \ (x<>y \ \text{and} \ y=z)$

e) $(x=y \ \text{or} \ y<>z) \ \text{or} \ (x<>y \ \text{or} \ y=z)$

f) $(x=y \ \text{or} \ y<>z) \ \text{and} \ (x<>y \ \text{or} \ y=z)$

3. Variabilele de tip întreg **x** și **y**, inițial egale, memorează valoarea 100. Se consideră următoarea secvență de program:

Limbajul C++/C

```
if(x>y) x=10*y-8*x;
else y=10*x-8*y;
```

Limbajul Pascal

```
if x>y then x:=10*y-8*x
else y:=10*x-8*y;
```

În urma executării secvenței de program alăturate, diferența absolută dintre valorile celor două variabile este:

- a) -200 b) -100 c) 0 d) 1 e) 100 f) 200

4. Se consideră următoarele două secvențe:

Limbajul C++/C

```
while ..... do
{
a=a-1;      { cout<<"20";
a--;
cout<<"20"; }while(a>=1);
}
```

Limbajul Pascal

```
while ..... do      repeat
begin                write('20');
a:=a-1;              a:=a-1;
write('20');        until a<1;
end;
```

Variabila de tip întreg **a** are inițial valoarea 21. Cele două secvențe sunt echivalente dacă punctele de suspensie se înlocuiesc cu:

- a) a=0 b) a>0 c) a>=1 d) a>1 e) a<=1 f) a<1

- Valoarea returnată de apelul **f(5)** pentru funcția alăturată este:

a) 3 b) 13 c) 48 d) 96 e) 144 f) 162

6. Concatenarea a două șiruri de caractere se poate realiza cu funcția predefinită:

Limbajul C++/C a) **strconcat** b) **strcmp** c) **strlen** d) **strcat** e) **strst**
Limbajul Pascal f) **strlwr**

7. Se consideră un graf neorientat cu nodurile numerotate de la 1 la 5 și muchiile [1,2], [1,5], [2,3], [2,4], [2,5], [3,4], [4,5]. Numărul lanțurilor distincte de lungime 3 de la nodul 1 la 4 este:

a) 3 b) 4 c) 5 d) 6 e) 7 f) 8

8. Se consideră un arbore cu rădăcină, cu 2020 noduri. Numărul minim de frunze pe care îl poate avea arborele este:

a) 0 b) 1 c) 2 d) 1010 e) 2019 f) 2020

9. Utilizând metoda backtracking se generează toate numerele, de cel mult trei cifre, formate cu cifre distincte, care au suma cifrelor egală cu 7 și **nu** sunt divizibile cu 10. Astfel, se generează în această ordine numerele: 106, 124, 142, 16, 205, Folosind aceeași metodă se generează toate numerele naturale cu cifre distincte, care au suma cifrelor egală cu 9 și **nu** sunt divizibile cu 5. Al șaselea număr generat este:

a) 135 b) 153 c) 162 d) 207 e) 216 f) 234

10. Se consideră următoarea secvență de program în care toate variabilele sunt numere întregi:

Limbajul C++/C <pre>for (i=0; i<= 2020; i++) { =i+1; a[i]=t; t--; }</pre>	Limbajul Pascal <pre>for i:=0 to 2020 do begin t:=i+1; a[i]:=t; dec(t); end;</pre>
--	--

Suma elementelor tabloului **a** este:

a) 2020 b) 2021 c) 4040 d) 4041 e) 2041210 f) 2043231

11. Folosind metoda bulelor tabloul unidimensional (5,6,10,20,1) este ordonat crescător: (1,5,6,10,20). Numărul de parcurgeri necesare pentru a ordona crescător tabloul este:

a) 9 b) 8 c) 7 d) 6 e) 5 f) 4

12. Un număr $\overline{abc}$ se numește **excepțional** dacă $b=a^c$. Mulțimea numerelor **excepționale** conține un număr de valori egal cu:

a) 36 b) 29 c) 26 d) 15 e) 6 f) 5

13. Se consideră următorul subprogram:

Limbaajul

C++/C

```
void f(int n)
{
    int i;
    if(n>0) for(i=1;i<=n;i++)
    { f(n-2);
      cout<<i<<' ';
      |printf("%d ",i);
    }
}
```

Limbaajul Pascal

```
procedure f (n:integer);
var i:integer;
begin
    if n>0 then
        for i:=1 to n do
            begin
                f(n-2);write(i,' ')
            end
        end;
end;
```

Valoarea lui n pentru care sunt afișate valorile 1 1 1 2 1 3 la apelul $f(n)$ este:

a) 12 b) 9 c) 6 d) 5 e) 4 f) 3

14. Variabila a memorează elementele unui tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 1 la 5, iar celelalte variabile sunt de tip întreg. Specificați care va fi conținutul variabilei a în urma executării secvenței de program date, dacă tabloul bidimensional are inițial conținutul alăturat:

1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

Limbaajul C++/C

```
for(i=1; i<=n; i++)
if(i<=n/2)
for(j=1; j<=i; j++)
{aux=a[i][j];
a[i][j]=a[i][n-j+1];
a[i][n-j+1]=aux;
}
else
for(j=1; j<=n-i+1; j++)
{ aux=a[i][j];
a[i][j]=a[i][n-j+1];
a[i][n-j+1]=aux;
}
```

Limbaajul Pascal

```
for i:=1 to n do
    if i<=n div 2 then for j:=1 to i do
        begin
            aux:=a[i,j]; a[i,j]:=a[i,n-j+1];
            a[i,n-j+1]:=aux;
        end
    else for j:=1 to n-i+1 do
        begin
            aux:=a[i,j]; a[i,j]:=a[i,n-j+1];
            a[i,n-j+1]:=aux;
        end;
```

a)	b)	c)	d)	e)	f)
5 4 3 2 1	5 4 3 2 1	5 2 3 4 1	1 2 5 4 3	1 2 3 5 4	1 2 3 4 5
5 2 3 4 1	1 2 3 4 5	5 4 3 2 1	1 2 5 4 3	1 2 3 5 4	1 2 3 4 5
5 2 3 4 1	1 2 3 4 5	5 4 3 2 1	1 2 5 4 3	1 2 3 5 4	1 2 3 4 5
5 2 3 4 1	1 2 3 4 5	5 4 3 2 1	1 2 5 4 3	1 2 3 5 4	1 2 3 4 5
5 4 3 2 1	5 4 3 2 1	5 2 3 4 1	1 2 5 4 3	1 2 3 5 4	1 2 3 4 5

15. Se consideră următorul subprogram:

Limbaajul C++/C

```
int T(int n)
{
```

Limbaajul Pascal

```
function T(n:integer):integer;
var o,m,c:integer;
```

<pre> int o,m=n,c=1; o=n; while(o>9) { c=c*10; o=o/10; } o=n%c*10+n/c; while(o!=n) { if(m<o) m=o; o=o%c*10+o/c; } return m; } </pre>	<pre> begin m:=n; c:=1;o:=n; while(o>9) do begin c:=c*10; o:=o div 10; end; o:=n mod c*10+n div c; while o<>n do begin if m<o then m:=o; o:=o mod c*10 + o div c; end; T:=m; end; </pre>
--	--

Știind că parametrul formal **n** este un număr natural format din 3 cifre, subprogramul **T** poate returna un număr de valori cu cifra sutelor 9 egal cu:

- a) 100** **b) 200** **c) 225** **d) 252** **e) 260** **f) 261**

Varianta 37

1. Se consideră următoarea secvență de program:

Limbajul C++/C

```
char c='7';
float a= c- '9';
cout<<a; | printf("%.0f",a);
```

Limbajul Pascal

```
var c:char; a:real;
c:='7';
a:=ord(c)- ord('9');
write(a:1:0);
```

Valoarea afișată în urma executării secvenței de program alăturate este:

- a) 79 b) -2 c) 2.0 d) 2 e) '79' f) -16

2. Se consideră următoarea listă de descendenți asociată unui arbore cu rădăcină cu 8 noduri:

```
1: 4,7,6,2
2: -
3: 4,6,5,2,7,8,1
4: -
5: -
6: -
7: 2
8: 7,2,4,1,6
```

Varianta care reprezintă **vectorul de tați** asociat acestui arbore este:

- a) 2 3 8 1 3 1 8 0 b) 8 7 0 1 8 1 1 3
c) 8 7 0 1 3 1 1 3 d) 0 3 1 8 3 1 1 1
e) 8 7 8 1 3 1 1 0 f) 0 7 8 3 1 3 3 1

3. În matricea de adiacență asociată unui graf neorientat cu n noduri, numărul de cifre de 1 aflate sub diagonala principală este egal cu $n \cdot (n-1) / 2$. Numărul de muchii ce trebuie adăugate la acest graf astfel încât acesta să devină complet este:

- a) $n-1$ b) n c) 1 d) $(n-1) / 2$ e) $n/2$ f) 0

4. Se consideră un graf neorientat cu 3675 de noduri și 10589 muchii. Gradul maxim pe care îl poate avea un nod din reprezentarea grafului ce conține un număr maxim de noduri izolate este:

- a) 147 b) 148 c) 146 d) 3666 e) 3674 f) 145

5. Șirul de caractere s ce desemnează o propoziție cu exact 11 cuvinte formate doar din litere mici, mari și separate prin câte un spațiu. Se consideră următoarea secvență de program:

Limbajul C++/C

```
int n;
char s[100], *p, c[100];
strcpy(s,s+(strchr(s,' ')-s));
p=strtok(s," ");
while (p && n)
{
    p=strtok(NULL," ");
    strcpy(c, p);
    n--;
}
```

Limbajul Pascal

```
var s,c,p:string[100];n:integer;
delete(s,1,pos(' ',s));
while n<>0 do
begin
    delete(s,1,pos(' ',s));
    c:=copy(s,1,pos(' ',s)-1);
    n:=n-1
end;
```

Pentru a memora în variabila **c** cuvântul din mijloc, valoarea atribuită variabilei **n** este:

- a) 11 b) 6 c) 5 d) 7 e) 3 f) 4

6. Se consideră şirul: 1,1,2,1,2,3,1,2,3,4,1,2,3,4,5,1,2,3,4,5,6,1,2,3,4,5,6,7... şi următoarea secvenţă de program:

Limbajul C++/C

```
int n,k,s=0;
cin>>n; | scanf("%d",&n);
k=1;
while(s<n)
    { s=s+k;k++; }
```

Limbajul Pascal

```
var n,k,s:integer;
read(n); k:=1; s:=0;
while s<n do
    begin
        s:=s+k; inc(k)
    end;
```

Expresia care determină termenul de pe o anumită poziţie **n** dată de la tastatură, dacă numerotarea termenilor pleacă de la valoarea 1 este:

- a) $s - (k - n) + 1$ b) $k - s + n - 1$ c) $k - s + n$ d) $k + s - n$ e) $n - k$ f) $k + n$

7. Se consideră următoarea secvenţă de program:

Limbajul C++/C

```
int i, j, n, a[10][10];
cin>>n; | scanf("%d",&n);
for (i=1;i<=n;i++)
    for (j=1;j<=n-i+1;j++)
    {
        a[i][j]=i+j;
        a[n-j+1][n+1-i]=i+j;
    }
```

Limbajul Pascal

```
var a: array [1..10, 1..10] of integer;
i, j, n: byte;
read(n);
for i:=1 to n do
    for j:=1 to n-i+1 do
    begin
        a[i][j]:= i+j;
        a[n-j+1][n+1-i]:=i+j
    end;
```

În urma executării secvenţei de program alăturate se obţine:

- a) un tablou bidimensional cu elementele simetrice faţă de diagonala principală dar nu şi faţă de diagonala secundară;
 b) un tablou bidimensional cu elementele simetrice faţă de diagonala secundară dar nu şi faţă de diagonala principală;
 c) un tablou bidimensional cu elementele simetrice atât faţă de diagonala principală cât şi faţă de diagonala secundară;
 d) un tablou bidimensional cu elementele identice pe coloane;
 e) un tablou bidimensional cu elementele identice pe linii;
 f) un tablou bidimensional cu toate elementele egale între ele.

8. Variabilele **a, b, i** şi **d** memorează numere naturale. Se consideră următoarea secvenţă de program:

Limbajul C++/C

```
for(i=a*b;i>=b;i--)
    if(i%a==0 && i%b==0)
        d=i;
cout<<d; | printf("%d",d);
```

Limbajul Pascal

```
for i:= a*b downto b do
    if i mod a=0 and i mod b=0 then
        d:=i;
write(d);
```

Valoarea afişată în urma executării secvenţei de program alăturate reprezintă:

- a) cel mai mare divizor comun b) numărul de multipli comuni
 c) cel mai mic multiplu comun d) numărul de divizori comuni

e) cel mai mare multiplu comun

f) numărul de divizori al produsului $a*b$

9. Se consideră următoarea secvență de program:

Limbajul C++/C

```
int a,i,c;
cin>>a; | scanf("%d",&a);
c=0;
for (i=1;i<=a;i++)
    if (i%5==0)
        { int j=i;
          while (j%5==0)
              { c++; j=j/5; }
        }
cout<<c; | printf("%d",c);
```

Limbaju Pascal

```
var a,i,c,j:integer;
read(a); c:=0;
for i:=1 to a do
    if i mod 5=0 then
        begin
            j:=i;
            while j mod 5=0 do
                begin
                    c:=c+1; j:=j div 5
                end
            end;
        end;
write(c);
```

Valoarea afișată în urma executării secvenței de program alăturate reprezintă:

- a) factorialul numărului a ;
- b) numărul cifrelor cu valoarea 0 de la sfârșitul factorialului numărului a ;
- c) puterea lui 5 din factorialul numărului a ;
- d) atât puterea lui 5 din factorialul numărului a , cât și numărul cifrelor cu valoarea 0 de la sfârșitul acestui factorial;
- e) numărul de elemente divizibile cu 5 mai mici decât a ;
- f) numărul de elemente divizibile cu 10 mai mici decât a .

10. Se consideră următoarea secvență de program:

Limbajul C++/C

```
struct oras {
    char strada[101];
    unsigned nr,cod_postal;
};
struct colet {
    char destinatar[51];
    struct oras adresa;
};
struct colet v[100];
```

Limbajul Pascal

```
type oras= record
    strada: string[101];
    nr, cod_postal: word
end;
colet= record
    destinatar: string[50];
    adresa: oras
end;
var v: array [1..101] of colet;
```

Varianta care reprezintă o accesare corectă a unei litere din numele unei străzi corespunzătoare unui colet transmis de o anumită firmă de curierat este:

- a) $v[5].adresa.oras[1]$
- b) $v[5].adresa[1].strada$
- c) $v[5].adresa.strada[1]$
- d) $v.colet.strada[5]$
- e) $adresa.v[5].strada[1]$
- f) $v.strada[1].adresa$

11. Fișierul **examen.txt** conține pe prima linie a sa valoarea unui număr natural n mai mic decât 100, iar pe următoarea linie n valori întregi separate prin câte un spațiu. Se consideră următoarea secvență de program:

Limbajul C++/C

```
ifstream f("examen.txt"); |
FILE *f; f= fopen("examen.txt","r");
```

Limbajul Pascal

```
var f,g:text; n,i:byte;
v: array [1..100] of integer;
```

```

int n, i, v[100];
f>>n; |fscanf(f,"%d",&n);
for (i=1;i<=n;i++) f>>v[i];
        |fscanf(f,"%d",&v[i]);
f.close();|fclose(f);
ifstream g("examen.txt"); |
FILE *g; g= fopen("examen.txt","r");
for (i=2;i<=n;i++) g>>v[i];
        |fscanf(g,"%d",&v[i]);
g.close(); | fclose(g);
cout<<v[n]; |printf("%d",v[n]);

```

```

assign(f,'examen.txt');
reset(f);
readln(f,n);
for i:= 1 to n do
    read(f,v[i]);
close(f);
assign(g,'examen.txt');
reset(g);
for i:= 2 to n do
    read(g,v[i]);
close(g); write(v[n]);

```

Valoarea afișată în urma executării secvenței de program alăturate reprezintă:

- | | |
|--------------------------------------|----------------------------------|
| a) prima valoare din fișier; | b) penultima valoare din fișier; |
| c) antepenultima valoare din fișier; | d) ultima valoare din fișier; |
| e) numărul de valori din fișier; | f) a doua valoare din fișier. |

12. Se consideră următoarea secvență de program:

Limbajul C++/C

```

unsigned n;
int c;
float f(int n)
{ if (n)
  { c++; return (n%10+ f(n/10)); }
  else return 0;
}

```

Limbajul Pascal

```

var n:word; c:integer;
function f(n: integer): real;
begin
  if n<>0 then
    begin
      inc(c);
      f:= n mod 10+ f(n div 10)
    end
    else f:= 0
end;

```

Apelul corect al funcției care returnează media aritmetică a cifrelor numărului natural **n** este:

- a) $f(n)/c$ b) $f(c)$ c) $f(n/c)$ d) $f(n)$ e) $f(c)/n$ f) $f(c/n)$

13. Se consideră următoarea secvență de program:

Limbajul C++/C

```

char s[101]="Sebastian
Nicholas", p[50]="bytes to mb";
strcpy(s+ (strchr(s,'a')+1 -s),
s+ strlen(s)-1);
s[3]++;
strncpy(s+3,p,2);
cout<<s<<endl;
|printf("%s\n",s);

```

Limbajul Pascal

```

var s, p: string[100];
s:='Sebastian Nicholas';
p:='bytes to mb';
delete(s,pos('a',s)
+ 2,length(s)-1);
s[4]:=chr(ord(s[4])-1);
delete(s, length(s)-1,2);
s:=s+ p[1]+ p[2];
writeln(s);

```

În urma executării secvenței de program alăturate se afișează:

- a) Sebyy b) Sebaby c) Nicholas d) Sebaty e) Sebby f) Seba

14. Variabila **n** memorează un număr natural și variabila **a** memorează un tablou bidimensional pătratic cu **n** linii și **n** coloane numerotate de la 1 la **n**. Se consideră următoarea secvență de program:

Limbajul C++/C

Limbajul Pascal


```

for (k=1;k<=n/2+1;k++){
    for (j=k;j<=n-k+1;j++)
        cout<<a[k][j]<<' ';
        | printf("%d ",a[k][j]);
    for (i=k+1;i<=n-k+1;i++)
        cout<<a[i][n-k+1]<<' ';
        | printf("%d ",a[i][n-k+1]);
    for (j=n-k;j>=k;j--)
        cout<<a[n-k+1][j]<<' ';
        | printf("%d ",a[n-k+1][j]);
    for (i=n-k;i>k;i--)
        cout<<a[i][k]<<' ';
        | printf("%d ",a[i][k]);
}

```

```

for k:= 1 to n div 2+ 1 do
begin
    for j:=k to n-k+1 do
        write(a[k,j], ' ');
    for i:=k+1 to n-k+1 do
        write(a[i,n-k+1], ' ');
    for j:=n-k downto k do
        write(a[n-k+1][j], ' ');
    for i:=n-k downto k+1 do
        write(a[i,k], ' ');
end;

```

În urma executării secvenței de program alăturate se vor afișa:

- a) elementele tabloului pe coloane, de la ultima la prima coloană;
- b) elementele tabloului în spirală;
- c) elementele tabloului pe linii, de la prima la ultima linie;
- d) elementele tabloului pe diagonale;
- e) elementele tabloului aflate pe coloane impare;
- f) elementele tabloului ce **nu** se află pe vreuna din cele două diagonale.

15. Nicholas are la Informatică un număr de **m** note stocate în tabloul unidimensional **note**, iar în variabila **teza** este trecut rezultatul obținut de el la lucrarea de sfârșit de semestru. Se știe că media se încheie cu un număr de **n** note (**m<n**), iar Nicholas dorește sa obțină media finală **x**.

Folosind metoda backtracking, Nicholas a creat un program care îi generează în tabloul unidimensional **note**, în continuarea celor **m** note existente, restul de **m-n** note necesare încheierii mediei. S-a notat cu **k** poziția pe care se generează pe rând restul notelor.

În rezolvarea programului s-a utilizat funcția **medie** al cărei apel calculează media curentă. Se consideră următoarea secvență de program:

Limbajul C++/C

```

float note[10], teza;
int n,m,k,as,ev,i,x;
float medie(float note[10],int m)
{
    float s=0;
    for (i=1;i<= m;i++)
        s=s+note[i];
    return (3*s+m*teza)/(4*m);
}
valid(float note[10],int k,int ev)
| void valid(float note[10],
int k,int *ev)
void
{
    ev=1; |*ev=1;
    if (k>m+1)
        if (note[k]< note[k-1])
            ev=0; |*ev=0;
    if (k==n)
        if (...)

```

Limbajul Pascal

```

type sir= array [1..11] of byte;
var note: sir;
    m,n,teza,k,i x:word;
    as,ev:boolean;
function
medie(note:sir;m:word):real;
var s:word;
begin
    s:=0;
    for i:=1 to m do s:=s+note[i];
    medie:=(3*s+m*teza)/(4*m)
end;
procedure valid( note:sir;k word;var
ev:boolean);
begin
    ev:=true;
    if k>m+1 then
        if note[k]< note[k-1] then
            ev:=false;

```

<pre> ev=0; *ev=0; } </pre>	<pre> if k=n then if ... then ev:=false end; </pre>
--	---

Pentru ca programul să genereze toate combinațiile de note de care Nicholas are nevoie pentru a obține media dorită, expresia corespunzătoare punctelor de suspensie din secvența de program este:

- a) **Limbajul C++/C:** `medie (note, n) != x;`
Limbajul Pascal: `medie (note, n) <> x`
- b) **Limbajul C++/C:** `medie (note, n) <= (x-0.5) && medie (note, n) > (x+0.5)`
Limbajul Pascal: `medie (note, n) <= (x-0.5) and medie (note, n) > (x+0.5)`
- c) **Limbajul C++/C:** `medie (note, n) < (x-0.5) || medie (note, n) > (x+ 0.5)`
Limbajul Pascal: `medie (note, n) < (x-0.5) or medie (note, n) > (x+ 0.5)`
- d) **Limbajul C++/C:** `medie (note, n) >= (x-0.5) && medie (note, n) <= (x+0.5)`
Limbajul Pascal: `medie (note, n) >= (x-0.5) and medie (note, n) <= (x+0.5)`
- e) **Limbajul C++/C:** `medie (note, n) < (x-0.5) || medie (note, n) >= (x+0.5)`
Limbajul Pascal: `medie (note, n) < (x-0.5) or medie (note, n) >= (x+0.5)`
- f) **Limbajul C++/C:** `medie (note, n) > (m-n)`
Limbajul Pascal: `medie (note, n) > (m-n)`

Varianța 38

1. Se consideră **A**, o mulțime de numere naturale. Cardinalul minim al acestei mulțimi, dacă o partiționăm în 5 partiții, iar numerele de elemente ale acestor partiții reprezintă termeni impari diferiți ai șirului lui Fibonacci este:

a) 12 b) 43 c) 20 d) 55 e) 6 f) 43

2. Se consideră un arbore cu rădăcină având următoarele caracteristici:

- numărul de noduri este 12;
- înălțimea este 4;
- numărul de frunze este 6;
- lungimea celui mai lung lanț elementar este egală cu 6;
- numărul de noduri de grad 1 este 7.

Un posibil **vector de tați** asociat acestui arbore ar putea fi:

a) 8 9 5 1 8 5 1 0 4 5 1 4 b) 9 6 10 1 0 9 1 6 5 1 2 10
 c) 4 1 8 0 1 2 5 1 5 12 2 5 d) 0 1 1 1 2 5 5 4 6 5 4 4
 e) 7 6 5 7 0 1 5 3 7 3 8 11 f) 12 4 6 0 1 2 12 2 1 7 6 4

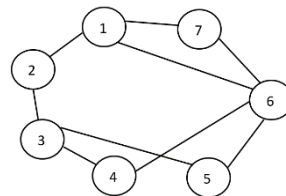
3. Se consideră o parolă de șase caractere, **nu** neapărat distincte, formată doar din litere mici și mari ale alfabetului englez (52 de caractere) și cifre. Numărul minim, respectiv maxim de încercări pentru a identifica respectiva parolă este:

a) 6 62 b) 1 6⁶² c) 1 6⁶¹ d) 1 62⁶ e) 6 62 f) 6 52

4.

Numărul de muchii care trebuie mutate din graful neorientat hamiltonian alăturat astfel încât acesta să devină eulerian, dar să rămână și hamiltonian este:

a) 0 b) 1 c) 2 d) 3 e) 4 f) 5



5. Variabilele **a** și **d** memorează numere naturale. Se consideră următoarea secvență de program:

Limbajul C++/C
`cin>>a; | scanf ("%d",&a);
 d=0;
 for (i=-a;i<=a;i++)
 if (a%i==0) d++;`

Limbajul Pascal
`read(a);
 d:=0;
 for i:=-a to a do
 if a mod i=0 then d:=d+1;`

În urma executării secvenței de program alăturate valoarea variabilei **d** este:

- a) numărul de divizori pozitivi ai lui **a**; b) numărul de divizori pozitivi și negativi ai lui **a**;
 c) numărul de divizori negativi ai lui **a**; d) cel mai mare divizor al lui **a**;
 e) numărul de divizori proprii ai lui **a**; f) valoarea variabilei **d** **nu** va putea fi calculată.

6. Fișierul **date.in** conține următoarele numere:

10
 1 2 3 4 5 6 7 8 9 10

Se consideră următoarea secvență de program în care tabloul unidimensional v a fost declarat parametru global:

Limbajul C++/C

```
int i, n, v[100];
ifstream f("date.in");
FILE *f,*g; f=fopen("date.in","r");

f>>n; | fscanf(f,"%d",&n);
for (i=1;i<=n;i=i+2)
    f>>v[i]; | fscanf(f,"%d",&v[i]);
f.close();|fclose(f);
ofstream g("date.in"); |
g=fopen("date.in","w");
g<<v[8];| fprintf(g,"%d",v[8]);
g.close();|fclose(g);
```

Limbajul Pascal

```
var f:text; i,n:byte;
v:array [1..100] of integer;

assign(f,'date.in');reset(f);
readln(f,n);
for i:= 1 to n do
    if i mod 2<>0then
        read(f,v[i]);
close(f);rewrite(f);
write(f,v[8]);
close(f);
```

Conținutul fișierului după executarea secvenței de program alăturate este:

- a) 8 b) o valoare reziduală c) 1 d) 8 9 10 e) 10 f) 0

7. Se consideră următoarea secvență de program:

Limbajul C++/C

```
typedef int sir[5];
sir v[100];int i,j;

for (i=1;i<=4;i++)
    for (j=1;j<=3;j++) v[i][j]=i+j;
for (i=1;i<=4;i++)
    for (j=1;j<=3;j++)
        if (j==3)
            cout<<v[i][j]<<endl;
            |printf("%d\n",v[i][j]);
            else cout<<v[i][j]<<' ';
            |printf("%d ",v[i][j]);
```

Limbajul Pascal

```
type sir=array [1..5] of
integer;
var v:array [1..100] of sir;
    i,j:integer;

for i:=1 to 4 do
    for j:=1 to 3 do v[i][j]:=i+j;
for i:=1 to 4 do
    for j:=1 to 3 do
        if j=3 then writeln(v[i][j])
        else write(v[i][j],' ');
```

În urma executării secvenței de program alăturate se afișează:

- | a) | b) | c) | d) | e) | f) |
|----------------|---------|---------|-------|-------|-----|
| 1 2 3 4 5 6 | 2 3 4 5 | 2 3 4 5 | 2 3 4 | 3 4 5 | 2 3 |
| 7 8 9 10 11 12 | 3 4 5 6 | 3 4 5 6 | 3 4 5 | 4 5 6 | 4 5 |
| | 4 5 6 7 | 4 5 6 7 | 4 5 6 | 5 6 7 | 6 7 |
| | 5 6 7 8 | | 5 6 7 | 6 7 8 | 8 9 |

8. Se consideră următoarea secvență de program:

Limbajul C++/C

```
void f(int k,int p)
{
    if (k*p>=0)
        if (p!=0)
        { ...
            cout<<k<<"* "<<p<<"=" ;
            cout <<k*p<<endl;
            |printf("%d* %d=%d\n",k,p,k*p);
        }
}
```

Limbajul Pascal

```
procedure f(k,p:integer);
begin
    if k*p>=0 then
        begin
            if p<>0 then
                begin
                    ...
                    writeln(k,'* ',p,'=' ,k*p)
                end
            end
        end
end
```

```

else f(k-1,10);
}

```

```

else f(k-1,10);
end
end;

```

Pentru a obține afișarea tablei înmulțirii de la 0 la 10 în urma apelului $f(10,10)$, apelul corespunzător punctelor de suspensie din secvența de program alăturată este:

- a) $f(k,p)$ b) $f(10,p)$ c) $f(k,1)$ d) $f(k-1,p-1)$ e) $f(k,p-1)$ f) $f(k-1,p)$

9. Variabila s memorează un șir de caractere. Se consideră următoarea secvență de program:

Limbajul C++/C

```

char s[101], cuv[5]="test";
while (strstr(s,cuv))
    strcpy(s+(strstr(s,cuv)-s),
           s+(strstr(s,cuv)-s+strlen(cuv)));

```

Limbajul Pascal

```

var s:string[101];
cuv:string[5];
cuv:='test';
while pos(cuv,s)<>0 do
    delete(s,pos(cuv,s),length(cuv));

```

În urma executării secvenței de program alăturate se realizează:

- a) eliminarea tuturor subșirurilor **test**; b) eliminarea ultimului subșir **test**;
c) dublarea tuturor subșirurilor **test**; d) eliminarea primului subșir **test**;
e) dublarea ultimei apariții a subșirului **test**; f) dublarea primei apariții a subșirului **test**.

10. Variabilele m, n, i, j, p, s și k memorează numere naturale. Tablourile bidimensionale $A_{n \times m}$ (n linii și m coloane), $B_{m \times p}$ (m linii și p coloane) memorează numere naturale. Numerotarea liniilor și a coloanelor începe cu valoarea 1.

Se consideră următoarea secvență de program:

Limbajul C++/C

```

for (i=1;i<=n;i++)
    for (j=1;j<=p;j++)
    {
        s=0;
        for (k=1;k<=m;k++)
            s=s+A[i][k]*B[k][j];
        cout<<s<<' ';
        | printf("%d ",s);
        if (j==p)
            cout<<endl;
        | printf("\n");
    }

```

Limbajul Pascal

```

for i:=1 to n do
    for j:=1 to p do
        begin
            s:=0;
            for k:=1 to m do
                s:=s+A[i][k]*B[k][j];
            write(s,' ');
            if j=p then
                writeln
        end;

```

În urma executării secvenței de program alăturate se afișează:

- a) tabloul bidimensional sumă dintre A și B ;
b) transpusa tabloului bidimensional A ;
c) tabloul bidimensional produs dintre A și B ;
d) tabloul bidimensional diferență dintre A și B ;
e) suma elementelor de pe ambele diagonale ale celor două tablouri bidimensionale;
f) produsul elementelor de pe ambele diagonale ale celor două tablouri bidimensionale.

11. Se consideră următoarea secvență de program:

Limbajul C++/C

```
char c;  
for(c='m';c<='r';c++)  
cout<<char(c- 5);  
|printf("%c",c-5);
```

Șirul de caractere afișat în urma executării secvenței de program alăturate este:

a) 104 b) mnopqr c) abcdef d) 109 e) hijklm f) 104105106107108109

Limbajul Pascal

```
var c:char;  
for c:= 'm' to 'r' do  
write(chr(ord(c)- 5));
```

12. Se consideră următoarea secvență de program:

Limbajul C++/C

```
char s[101];int k, p, c;  
cin.getline(s,101); |gets(s);  
k=0;p=strlen(s)-1;  
while (k!=strlen(s))  
{ c=s[k]+s[p];  
s[k]=c-s[k];  
s[p]=c-s[k];  
k++;p--;  
}  
cout<<s; |printf("%s",s);
```

Limbajul Pascal

```
var s: string[100]; k,p,c: integer;  
readln(s);  
k:=1;p:=length(s);  
while (k<= length(s)) do  
begin  
c:=ord(s[k])+ord(s[p]);  
s[k]:=chr(c-ord(s[k]));  
s[p]:=chr(c-ord(s[k]));  
k:=k+1;p:=p-1  
end;  
write(s);
```

În urma executării secvenței de program alăturate se afișează:

- a) șirul de caractere dat de la tastatură;
- b) șirul de caractere dat de la tastatură, răsturnat;
- c) șirul de caractere fără cele de pe pozițiile **k** și **p**;
- d) subșirul de caractere aflate între pozițiile **k** și **p**;
- e) toate caracterele din șirul **s** care **nu** se află între pozițiile **k** și **p**;
- f) primul și ultimul caracter din **s**.

13. Fișierul **date.in** conține informații despre trei elevi. Pentru fiecare elev sunt precizate următoarele: numele; cinci note și teza pentru o anumită materie:

Ana

7 5 8 3 6

6

Sebby

10 9 10 9 10

10

Dan

9 8 9 9 7

9

Se consideră următoarea secvență de program:

Limbajul C++/C

```
ifstream f("date.in");|  
FILE *f; f= fopen("date.in","r");  
struct materie  
{ char nume[51]; unsigned note[6];  
float teza;  
} clasa[30];  
float s;int i,j;
```

Limbajul Pascal

```
type sir=array [1..6] of byte;  
materie= record  
nume: string[50];  
note: sir; teza: real  
end;  
var clasa:array [1..30] of  
materie;  
s:real; i,j:integer; f:text;
```

```

s=0;
for (j=1;j<=3;j++)
{ f>>clasa[j].nume; |
  fscanf(f,"%s",clasa[j].nume);
  for (i=1;i<= 5;i++)
  { f>>clasa[j].note[i]; |
    fscanf(f,"%d",&clasa[j].note[i]);
    s=s+clasa[j].note[i];
  }
  f>>clasa[j].teza; |
  fscanf(f,"%f",&clasa[j].teza);
}
s=s-71;
cout<<(3*s+5*clasa[2].teza)/20; |
printf("%f", (3*s
+5*clasa[2].teza)/20);
f.close();|fclose(f);

```

```

assign(f,'date.in');reset(f);
s:=0;
for j:=1 to 3 do
  begin
    readln(f,clasa[j].nume);
    for i:=1 to 5 do
      begin
        read(f,clasa[j].note[i]);
        s:=s+clasa[j].note[i]
      end;
    readln(f,clasa[j].teza)
  end;
s:=s-71;
write((3*s+5*clasa[2].teza)/20);
close(f);

```

În urma executării secvenței de program alăturate se afișează:

- a) media celor trei elevi la respectiva materie; b) suma mediilor celor trei elevi;
 c) media lui **Dan** la respectiva materie; d) media lui **Sebby** la respectiva materie;
 e) cea mai mare medie dintre cele trei; f) cea mai mică medie dintre cele trei.

14. Se consideră o parolă cu n caractere ce este alcătuită din cifre dispuse în progresie aritmetică. Folosind metoda backtracking se generează șiruri de numere de lungime n până la depistarea parolei respective. Se consideră următoarea secvență de program:

Limbajul C++/C

```

void valid(int parola[10],int k,int &ev) |
void valid(int parola[10], int k, int *ev)
{   ev= 1; |*ev= 1;
    if (k>2)
        if (...)
            ev= 0; |*ev= 0;
}

```

Limbajul Pascal

```

type sir= array [1..10] of integer;
procedure valid( parola: sir; k: word; var ev: boolean);
begin
  ev:= true;
  if k>2 then
    if ... then ev:= false
end;

```

Expresia corespunzătoare punctelor de suspensie din secvența de program alăturată pentru ca cifra de pe poziția k ($k > 2$) să fie considerată validă și prin urmare variabila **ev** să primească valoarea 1(C++/C)/True(Pascal) este:

Limbajul C++/C

- a) parola[k] - parola[k-1] != parola[2] - parola[1]
 b) parola[k] - parola[k-1] != parola[k+1]- parola[k]
 c) parola[k] <> parola[k-1]
 d) parola[k] - parola[k-1] >= parola[2] - parola[1]
 e) parola[k+1] - parola[k] != parola[2] - parola[1]

f) parola[k] == parola[2] - parola[1]

Limbajul Pascal

- a) parola[k] - parola[k-1] <> parola[2] - parola[1]
- b) parola[k] - parola[k-1] <> parola[k+1] - parola[k]
- c) parola[k] != parola[k-1]
- d) parola[k] - parola[k-1] >= parola[2] - parola[1]
- e) parola[k+1] - parola[k] <> parola[2] - parola[1]
- f) parola[k] = parola[2] - parola[1]

15. Variabilele n ($n \geq 2$) și i memorează numere naturale și tablou bidimensional pătratic a (n linii și n coloane) are valori din mulțimea $\{1, 2, 3, 4, 5\}$. Se consideră următoarea secvență de program:

Limbajul C++/C

```
for (i=n; i>= 1; i--)  
    if (i!=n-i+1)  
    {  
        a[i][n-i+1]=a[i][n-  
i+1]+a[i][i];  
        a[i][i]=a[i][n-i+1]-a[i][i];  
        a[i][n-i+1]=a[i][n-i+1]-  
a[i][i];  
    }
```

Limbajul Pascal

```
for i:=n downto 1 do  
    if i<>n-i+1 then  
        begin  
            a[i,n-i+1]:= a[i,n-i+1]+a[i,i];  
            a[i,i]:= a[i,n-i+1]-a[i,i];  
            a[i,n-i+1]:=a[i,n-i+1]-a[i,i]  
        end;
```

În urma executării secvenței de program alăturate se realizează:

- a) interschimbarea elementelor de pe linia i și coloana $n-i+1$;
- b) egalarea valorilor de pe cele două diagonale;
- c) interschimbarea elementelor de pe cele două diagonale;
- d) înlocuirea elementelor de pe diagonala principală cu cele de pe diagonala secundară;
- e) interschimbarea elementelor de pe coloana i și linia $n-i+1$;
- f) înlocuirea elementelor de pe diagonala secundară cu cele de pe diagonala principală.

Varianta 39

1. Fie un șir alcătuit din 100 de elemente numere naturale (componentele șirului se citesc de la tastatură prin intermediul variabilei întregi **a**). Următoarea secvență de cod determină, în variabila întreagă **nr**, numărul tuturor elementelor din șir care memorează un număr alcătuit din cel puțin două cifre. Stabiliți expresiile care pot înlocui punctele de suspensie.

Limbajul C++/C

```
nr=100;
for(i=1;i<=100;i++)
{cin>>a; | scanf("%d",&a);
if(9>=...)
nr=...+nr;}
```

Limbajul Pascal

```
nr:=100;
for i:=1 to 100 do begin
  readln(a);
  if 9>=... then
    nr:=...+nr;
end;
```

- | | | |
|-----------|------------|------------|
| a) a și i | b) a și -i | c) a și -1 |
| d) a și 1 | e) a și a | f) a și -a |

2. Variabila **i** memorează un număr întreg, iar **s** memorează un șir alcătuit din cel mult 20 de caractere. Rezultatul obținut, în urma rulării secvenței de mai jos, este:

Limbajul C++/C

```
strcpy(s,"VAPOARE");
i=0;
while(i<strlen(s)-1)
{if(strchr("AEIOU",s[i])!=0)
{ s[i]=s[i]+1;
strcpy(s+i+1,s+i+2);
}
i++;}
cout<<s; | printf("%s",s);
```

Limbajul Pascal

```
s:='VAPOARE' ;
i:=1;
while i<= length(s)-1 do
begin
if pos(s[i],'AEIOU')<>0 then
begin
s[i]:=succ(s[i]);
delete(s,i+1,1);
end;
i:=i+1;
end;
write(s);
```

- | | | |
|----------|----------|----------|
| a) VARE | b) VBPR | c) VBPRF |
| d) VBPRO | e) VBPRE | f) VPRBO |

3. După execuția următoarei secvențe, stabiliți numărul elementelor cu valoarea 9 din tabloul unidimensional **a**.

Limbajul C++/C

```
int a[] = {0, 1, 2, 3, 0, 4, 5, 6};
int i = 0, x = 9;
do{
a[i++] = x;
}
while(i<6&& a[i]);
```

Limbajul Pascal

```
type vector=array[1..8] of
integer;
var
a:vector = (0,1,2,3,0,4,5,6);
i,x:integer;
begin
```

```

i := 1;    x := 9;
repeat
    a[i] := x; i := i + 1;
until (i > 6) or (a[i] = 0);
end.

```

a) niciunul b) unul c) două d) trei e) patru f) cinci

4. În vederea sortării crescătoare a unui șir de valori întregi, folosind metoda bulelor (bubble sort), un program citește valorile următoare **2, 40, 17, 1, 51, 34, 20, 63** și le memorează într-un tablou unidimensional. După câte parcurgeri ale șirului, valoarea **40** ajunge pe locul final în tabloul unidimensional sortat crescător?

a) 0 b) 1 c) 2 d) 3 e) 4 f) 5

5. Într-un tablou bidimensional de dimensiuni **$n \times n$** , având liniile și coloanele numerotate de la **1** la **n** , condiția pentru ca elementul de pe linia **i** și coloana **j** să fie situat deasupra diagonalei principale și deasupra diagonalei secundare este:

Limbajul C++/C

a) $(i \leq j) \ \&\& \ (i + j < n)$	b) $(i < j) \ \&\& \ (i + j < n + 1)$	c) $(i < n) \ \&\& \ (i + j < n - 1)$
d) $(i < j) \ \ (i + j < n + 1)$	e) $(i \leq j) \ \&\& \ (i + j \leq n + 1)$	f) $i < n + j$

Limbajul Pascal

a) $(i \leq j) \ \text{AND} \ (i + j < n)$	b) $(i < j) \ \text{AND} \ (i + j < n + 1)$	c) $(i < n) \ \text{AND} \ (i + j < n - 1)$
d) $(i < j) \ \text{OR} \ (i + j < n + 1)$	e) $(i \leq j) \ \text{AND} \ (i + j \leq n + 1)$	f) $i < n + j$

6. Se consideră un graf neorientat cu **8** noduri și **28** de muchii. Indicați numărul minim de muchii care pot fi eliminate, astfel încât graful parțial obținut să conțină două componente conexe, cu cel puțin două noduri fiecare.

a) 4 b) 6 c) 8 d) 10 e) 12 f) 16

7. Variabilele **x** , **y** și **z** sunt de tip întreg și memorează numere naturale din intervalul **$[1, 10^3]$** . Indicați o expresie care are valoarea **1** în C++/C sau valoarea **TRUE** în Pascal, dacă și numai dacă valoarea variabilei **x** este strict mai mare decât valoarea oricăreia dintre variabilele **y** și **z** .

Limbajul C++/C

a) $x * y > y * z \ \&\& \ x * z > y * z$	b) $x * z > x * y \ \&\& \ y * z > y * x$
c) $y * z > x * z \ \&\& \ y * x > z * x$	d) $y * z > y * x \ \&\& \ y * z > z * x$
e) $x * y > y * z \ \ x * z > y * z$	f) $y * z > y * x \ \ y * z > z * x$

Limbajul Pascal

a) $(x * y > y * z) \ \text{AND} \ (x * z > y * z)$	b) $(x * z > x * y) \ \text{AND} \ (y * z > y * x)$
c) $(y * z > x * z) \ \text{AND} \ (y * x > z * x)$	d) $(y * z > y * x) \ \text{AND} \ (y * z > z * x)$
e) $(x * y > y * z) \ \text{OR} \ (x * z > y * z)$	f) $(y * z > y * x) \ \text{OR} \ (y * z > z * x)$

8. O variabilă întreagă x conține cel mai mic număr natural nenul, multiplu de 36, divizibil cu toate numerele prime mai mici decât 10. Indicați o expresie care are valoarea 1 în C++/C sau valoarea **TRUE** în Pascal.

Limbajul C++/C

- a) $(x < 1000) \&\& (x \% 27 == 0)$ b) $(x > 1000) \&\& ((x * x * x) \% 1000 == 0)$
 c) $((x * x) / 16) \% 2 == 0$ d) $(x \% 100 == 0) || (x / 100 == 0)$
 e) $(x \% 9 == 0) \&\& (x \% 25 == 0)$ f) $(x != 1260) || (x < 1200)$

Limbajul Pascal

- a) $(x < 1000) \text{ AND } (x \text{ MOD } 27 = 0)$ b) $(x > 1000) \text{ AND } ((x * x * x) \text{ MOD } 1000 = 0)$
 c) $((x * x) \text{ DIV } 16) \text{ MOD } 2 = 0$ d) $(x \text{ MOD } 100 = 0) \text{ OR } (x \text{ DIV } 100 = 0)$
 e) $(x \text{ MOD } 9 = 0) \text{ AND } (x \text{ MOD } 25 = 0)$ f) $(x <> 1260) \text{ OR } (x < 1200)$

9. Știind că variabila n reține un număr întreg strict pozitiv, indicați semnificația valorii variabilei întregi x după execuția următoarei secvențe:

Limbajul C++/C

```
x=1;
while(n>1)
{
  n=n/2; x++;
}
cout<<x; |
printf("%d",x);
```

Limbajul Pascal

```
x:=1;
while n>1 do
begin
  n:=n div 2;
  x:=x+1;
end;
write(x);
```

- a) suma puterilor din descompunerea în factori primi a numărului n b) puterea la care apare 2 în descompunerea în factori primi a numărului n
 c) numărul divizorilor pozitivi ai lui n d) $\lfloor \log_2 n \rfloor - 1$
 e) $\lfloor \log_2 n \rfloor$ f) $\lfloor \log_2 n \rfloor + 1$
 Notăție: $[a]$ partea întreagă a numărului real a

10. Fie f și g două subprograme având următoarele definiții. Precizați valoarea returnată de apelul $g(6)$.

Limbajul C++/C

```
int f(int x){
  if (x%2==0)
    return f(x/2);
  else return x;
}
int g(int x){
  if(x<1) return 1;
  else return f(x*g(x-1));
}
```

Limbajul Pascal

```
function
f(x:integer):integer;
begin
  if x mod 2=0
    then f:=f(x div 2)
    else f:= x;
end;
function
g(x:integer):integer;
begin
```

```

        if x<1 then g:=1
        else g:=f(x*g(x-1));
    end;

```

- a) 3 b) 9 c) 30 d) 45 e) 210 f) 315

11. Un graf neorientat este **eulerian** dacă:

- a) este conex și conține cel puțin un ciclu elementar b) este conex și nu conține cicluri elementar
 c) este conex și suma elementelor de pe fiecare coloană a matricei de adiacență este număr par d) matricea de adiacență este simetrică față de diagonala principală
 e) conține cel puțin un ciclu hamiltonian f) conține un singur ciclu elementar

12. În urma executării următorului program, valorile afișate pe ecran sunt:

Limbaajul C++

```

#include <iostream>
using namespace std;
int x,y;
void g(int &a,
int &b)
{a=a+5; b=b+a;}
int main()
{x=1; y=2;
g(y,x);
cout<<x<<" "<<y;
cout<<" ";
g(y,x);
cout<<x<<" "<<y;
return 0;}

```

Limbaajul C

```

#include <stdio.h>
int x,y;
void g(int *a,
int *b)
{*a=*a+5;
*b=*b+*a;}
int main()
{ x=1; y=2;
g(&y,&x);
printf("%d %d",x,y);
printf(" ");
g(&y,&x);
printf("%d %d",x,y);
return 0;}

```

Limbaajul Pascal

```

var x,y:integer;
procedure g(var
a,b: integer);
begin a:=a+5;
b:=b+a; end;
begin
x:=1; y:=2;
g(y,x);
write(x,' ',y);
write(' ');
g(y,x);
write(x,' ',y);
end.

```

- a) 8 7 20 12 b) 8 7 12 20 c) 7 8 12 20
 d) 7 8 20 12 e) 12 20 7 8 f) 12 20 8 7

13. Precizați numărul de șiruri distincte formate din exact o literă **A**, două litere **B**, trei litere **C** și patru litere **D**.

- a) 2500 b) 3600 c) 7560
 d) 10300 e) 12600 f) 151200

14. Fie arborele cu 8 noduri și cu muchiile [1,2], [1,3], [1,4], [4,5], [6,4], [1,8], [4,7]. Câți **vectori de tați** distincți se pot construi pentru acest arbore? Doi **vectori de tați** sunt distincți dacă există cel puțin o poziție pentru care elementele din respectivele poziții sunt distincte.

- a) 7 b) 8 c) 28 d) 36 e) 8 f) 40320
 !

15. Stabiliți rezultatul execuției secvenței de mai jos, unde variabilele **x** și **b** rețin numere naturale cunoscute ($1 \leq x \leq 1000$, $1 < b \leq 10$), iar **s** este o variabilă întreagă:

Limbajul C++/C

```
s=0;
while (x>0)
{
s=s+ x % b;
x=x / b;
}
if (s % (b-1)==0)
cout<<"da"; | printf("da");
else
cout<<"nu"; | printf("nu");
```

Limbajul Pascal

```
s:=0;
while x>0 do
begin
s:=s+ x mod b;
x:=x div b;
end;
if s mod (b-1)=0 then
write('da')
else write('nu');
```

a) verifică dacă suma cifrelor reprezentării în baza **b-1** a numărului **x** este divizibilă cu **b-1**

c) verifică dacă suma cifrelor reprezentării în baza **b** a numărului **x** este divizibilă cu **b**

e) verifică dacă suma cifrelor lui **x** este divizibilă cu **b-1**

b) verifică dacă numărul **x** este divizibil cu **b-1**

d) verifică dacă numărul **x** este divizibil cu **b**

f) niciuna dintre variantele anterioare

Varianta 40

1. Se consideră subprogramul **f** având definiția următoare. Stabiliți valoarea variabilei **n** de tip întreg știind că, la apelul **f(n)**, subprogramul returnează valoarea **2014**?

Limbajul C++/C

```
int f(int x)
{ if (x>=5)
    return f(x-1)+x;
  return 2*x;
}
```

Limbajul Pascal

```
function f(x:integer):integer;
begin
  if x<5 then f:=2*x
  else f:=f(x-1)+x;
end;
```

- a) 16 b) 30 c) 62 d) 63 e) 88 f) 100

2. În următoarea secvență de program, variabilele **k**, **i** și **j** sunt de tip întreg, iar variabila **A** memorează un tablou bidimensional cu 7 linii și 7 coloane (numerotate de la 1 la 7) cu elemente de tip întreg. Precizați care este cea mai mare valoare memorată în matricea **A** la finalul executării secvenței?

Limbajul C++/C

```
k=1;
for(i=1;i<=7;i++)
  for(j=1;j<=7;j++)
  { A[i][j]=k++;
    A[i][8-j]=k; }
```

Limbajul Pascal

```
k:=1;
for i:=1 to 7 do
  for j:=1 to 7 do
  begin
    A[i][j]:=k; k:=k+1;
    A[i][8-j]:=k;
  end;
```

- a) 0 b) 7 c) 10 d) 37 e) 50 f) 51

3. Se consideră următoarea secvență de cod. Identificați ce se va afișa dacă de la tastatură se vor introduce, în ordine, șirurile de caractere **student**, **carte** și **birou**:

Limbajul C++/C

```
char a[256], b[256]; int i;
strcpy(b, "");
for(i=0;i<3;i++)
{ cin>>a; | scanf("%s",a);
  strcat(b, a+i);
}
cout<<b; | printf("%s",b);
```

Limbajul Pascal

```
var a,b:string; i:integer;
begin
  b:='';
  for i:=0 to 2 do
  begin
    readln(a);
    b:=b+copy(a,i+1,length(a));
  end;
  write(b);
end.
```

- a) scb b) studencartbiro c) studentarterou
d) studentcartbir e) tudenartiro f) tudentrteou

4. Se consideră definite trei variabile întregi **x**, **y** și **z** și următoarele două expresii. Stabiliți afirmația adevărată.

Limbajul C++/C

```
p=!((x == y)&&(x == z));
q=(x!=y) || (x!=z);
```

Limbajul Pascal

```
p:=NOT((x = y)AND(x = z));
q:=(x <> y)OR(x <> z);
```

- a) p egal cu q dacă și numai dacă x egal cu y b) p egal cu q dacă și numai dacă x egal cu z
 c) oricare ar fi x, y, z p este diferit de q d) oricare ar fi x, y, z p este egal cu q
 e) există x, y, z astfel încât p diferit de q f) niciuna dintre variantele anterioare
5. Subprogramul **f** primește prin intermediul parametrului **n** un număr natural și returnează numărul maxim de cifre consecutive situate pe poziții alăturate în scrierea numărului **n**. De exemplu, la apelul **f(23523455)** subprogramul returnează numărul **4**. Determinați numărul de valori distincte **n** cu exact **4** cifre pentru care expresia **f(n)** are valoarea **3**.
 a) 63 b) 121 c) 130 d) 152 e) 160 f) 181
6. Un tip de date întreg pe **n** biți ($n > 1$, număr natural) va putea reține valori întregi din intervalul:
 a) $[-2^{n-1}, 2^{n-1}-1]$ b) $[-2^n, 2^n - 1]$ c) $[-2^n, 2^n]$
 d) $[0, 2^{n-1}]$ e) $[0, 2^n]$ f) $[0, 2^{n-1}-1]$
7. Precizați valoarea afișată în urma executării secvenței de program:
- | | |
|--|--|
| Limbajul C++/C
<pre>int i, p=16; while(1) { if (i=5) break; p+=i; i+=2; } cout<<p; printf("%d",p);</pre> | Limbajul Pascal
<pre>var i, p:integer; begin p:=16; i:=1; while TRUE do begin if i=1 then break; p:=p+i; i:=i+2; end; write(p); end.</pre> |
|--|--|
- a) 10 b) 16 c) 17
 d) 20 e) 24 f) instrucțiunea **while** rulează la infinit
8. Se consideră un tablou unidimensional **a** în care elementele sunt, în ordine: **1, 3, 5, 7, 10, 16, 21**. Pentru a afla poziția pe care se află valoarea **x=10** în tablou, se aplică metoda căutării binare. Identificați succesiunea corectă de elemente a căror valoare se compară cu valoarea **x**.
 a) 21, 16, 10 b) 7, 16, 10 c) 1, 3, 5, 7, 10
 d) 5, 7, 10 e) 7, 10 f) 10
9. Un tablou bidimensional cu 8 linii, format doar din elemente 0 și 1, are următoarele proprietăți:
 Prima linie conține un singur element cu valoarea 1;

Linia j conține de două ori mai multe valori nenule decât linia $j-1$, $2 \leq j \leq 8$;

Ultima linie conține un singur element cu valoare 0.

Determinați numărul total de elemente cu valoare 0 din tabloul bidimensional.

- a) 528 b) 600 c) 688 d) 769
e) 777 f) Nu există un tablou bidimensional cu aceste proprietăți

10. Tabloul unidimensional A conține, începând cu indicele 1, elementele: (1, 2, 2, 212, 12212, 21212212, 1221221212212, ...). Stabiliți numărul de cifre de 2 conținute de cel de-al 15-lea element al tabloului?

- a) 377 b) 391 c) 400
d) 520 e) 588 f) 610

11. Matricea de adiacență asociată unui graf neorientat cu 17 noduri are 40 elemente nenule. Numărul maxim de componente conexe în graf este:

- a) 9 b) 10 c) 11 d) 12 e) 13 f) 14

12. Fie un număr a care aparține intervalului [410, 681]. Pentru a verifica dacă a este prim, numărul minim de numere care trebuie testate pentru a fi divizori ai lui a este:

- a) 10 b) 13 c) 25 d) 205
e) 339 f) 640

13. Utilizând metoda backtracking se generează toate permutările mulțimii {1, 2, 3, 4, 5} în ordine lexicografică. Primele cinci soluții generate sunt 12345, 12354, 12435, 12453, 12534. Spunem că o permutare p a mulțimii {1, 2, 3, 4, 5} are numărul de ordine k , dacă este a k -a permutare generată astfel. Permutarea 12354 are numărul de ordine 2, iar permutarea 12534 are numărul de ordine 5. Precizați numărul de ordine al permutării 51423.

- a) 98 b) 99 c) 100
d) 101 e) 110 f) 111

14. Variabilele m și n rețin valori numere natural nenule. Precizați ce execută secvența de cod de mai jos, unde c , d și x sunt variabile întregi.

Limbajul C++/C

```
c=m; d=n;
while (c!=d)
if (c>d) c=c-d;
else d=d-c;
x=m*n/c;
cout<<x; | printf("%d",x);
```

Limbajul Pascal

```
c:=m; d:=n;
while c<>d do
if c>d then c:=c-d
else d:=d-c;
x:=m*n div c;
write(x);
```

A. Calculează și afișează cel mai mic multiplu comun al numerelor naturale nenule m și n

B. Dacă $m=9$ și $n=12$, atunci afișează $x=36$

C. Dacă $m=11$ și $n=6$, atunci afișează $x=17$

- a) A b) A,B c) A,B,C
d) B e) B,C f) C

15. Determinați numărul de grafuri neorientate cu mulțimea nodurilor $\{1, 2, 3, 4, 5, 6, 7, 8\}$ în care, atât nodurile 2 și 3, cât și nodurile 2 și 4 sunt neadiacente.

a) 4^4

b) 4^{10}

c) 2^{23}

d) $2^{24}-2$

e) 2^{25}

f) 4^{13}

Varianta 41

1. Numim graf complementar al unui graf neorientat G graful neorientat G_1 cu aceeași mulțime a nodurilor ca și G și cu proprietatea că două noduri sunt adiacente în G_1 dacă și numai dacă nu sunt adiacente în G . Dacă G are n noduri și m muchii, câte muchii are G_1 ?
- a) exact $n(n-1)/2-m$ b) exact $n-m$ c) exact $(n-1)/2$
d) minimum $n(n-1)/2-m$ e) minimum $n-m$ f) maximum $n(n-1)/2-m$

2. Cu ce expresie trebuie completată secvența lipsă (marcată prin...) din funcția următoare pentru ca $f(x, 2)$ să aibă ca rezultat suma exponenților factorilor primi ce intră în descompunerea lui x ?

Limbajul C++/C

```
int f(int x, int d)
{
    if(...) return 1;
    if(x%d==0) return 1+f(x/d,
d); return f(x, d+1);
}
```

Limbajul Pascal

```
function f(x,
d:integer):integer;
begin
    if ... then f:=1
    else
        if x mod d=0
            then f:= 1+f(x div d, d)
        else f:=f(x, d+1);
    end;
```

- a) $x==1$ (Limbajul C++/C) b) $x==0$ (Limbajul C++/C) c) $x<d/2$
a) $x=1$ (Limbajul Pascal) b) $x=0$ (Limbajul Pascal)
d) $x<d$ e) $x>d$ f) $x<=d$

3. Instrucțiunea care afișează cea mai din stânga poziție unde se află valoarea întreagă x , sau afișează -1 , dacă x nu apare în tabloul unidimensional a cu n elemente numere întregi, este:

Limbajul C++/C

```
1.
for(i=0;i<n&& a[i]==x;i++)
    if (i<n) cout<<i; |
        printf("%d",i);
    else cout<<-1; |
        printf("-1");
2.
for(i=0;i<n&& a[i]!=x;i++);
    if (i==n) cout<<i;
|printf("%d",i);
    else cout<<-1; |
        printf("-1");
3.
for(i=0;i<n&& a[i]==x;i++)
    if (i==n) cout<<i; |
        printf("%d",i);
    else cout<<-1; |
```

Limbajul Pascal

```
1.i:=0;
while (i<n)and(a[i]=x) do
begin
    if(i<n) then write(i)
        else write(-1);
    i:=i+1; end;
2. i:=0;
while (i<n) and (a[i]<>x) do
    i:=i+1;
if i=n then write(i)
    else write(-1);
3. i:=0;
while (i<n) and (a[i]=x) do
begin
    if i=n then write(i)
        else write(-1) ;
```

<pre> printf("-1"); 4. for(i=0;i<n&& a[i]!=x;i++); if (i<n) cout<<i; printf("%d",i); else cout<<-1; printf("-1"); 5. for(i=0;i==n&& a[i]!=x;i++); if (i==n) cout<<i; printf("%d",i); else cout<<-1; printf("-1"); 6. for (i=0;i<n;i++) if (a[i]==x)cout<<i; printf("%d",i); else cout<<-1; printf("-1"); </pre>	<pre> i:=i+1; end; 4. i:=0; while (i<n) and (a[i]<>x) do i:=i+1; if i < n then write(i) else write(-1); 5. i:=0; while (i=n) and (a[i]<>x) do i:=i+1; if i=n then write(i) else write(-1) ; 6. for i: = 0 to n-1 do if a[i] = x then write(i) else write(-1) ; </pre>
--	---

a) 1 b) 2 c) 3 d) 4 e) 5 f) 6

4. Apelul **F(7)** returnează:

Limbajul C++/C <pre> int F(int N) { if (N==0) return 1; return F(N-1) + F(N-1); } </pre>	Limbajul Pascal <pre> function F(N:integer):integer; begin if N=0 then F:= 1 else F:= F(N-1) + F(N-1); end; </pre>
a) 60 b) 64 c) 120	d) 128 e) 240 f) 256

5. Se consideră un șir de caractere **s**. Stabiliți rezultatul afișării pe ecran, în urma executării următoarei secvențe de program:

Limbajul C++/C <pre> char s[15]="ABCDEFGH",*p,x[15]; p=s; p+=4; strcpy(x,s+2); strcat(x,p); cout<<x; printf("%s",x); </pre>	Limbajul Pascal <pre> var s,p,x:string; begin s:= 'ABCDEFGH';p:=s; p:=copy(p,5,length(s)); x:=copy(s,3,length(s)); x:=x+p; write(x); end. </pre>
a) ABCEFG b) BCDEFGDEFG c) CDEFGGEFG	

d) CDEFGABCD

e) DEFGDEFG

f) EFGEFG

6. Precizați care dintre următoarele expresii are valoarea 1|TRUE dacă și numai dacă numărul natural n este divizibil cu 3 și are ultima cifră 4 sau 6.

Limbajul C++/C

A. $n/3==0 \&\& (n\%10==4 \mid \mid n\%10==6)$

B. $n\%3==0 \&\& (n\%10==4 \mid \mid n\%10==6)$

C. $(n\%3==0 \&\& n\%10==4) \mid \mid (n\%3==0 \&\& n\%10==6)$

D. $(n\%3==0 \&\& n\%10==4) \mid \mid n\%10==6$

Limbajul Pascal

A. $(n \text{ DIV } 3=0) \text{ AND } ((n \text{ MOD } 10=4) \text{ OR } (n \text{ MOD } 10=6))$

B. $(n \text{ MOD } 3=0) \text{ AND } ((n \text{ MOD } 10=4) \text{ OR } (n \text{ MOD } 10=6))$

C. $((n \text{ MOD } 3=0) \text{ AND } (n \text{ MOD } 10=4)) \text{ OR } ((n \text{ MOD } 3=0) \text{ AND } (n \text{ MOD } 10=6))$

D. $((n \text{ MOD } 3=0) \text{ AND } (n \text{ MOD } 10=4)) \text{ OR } (n \text{ MOD } 10=6)$

a) b) B

c) A,B

d) B,C

e) B,D

f) D

7. Se generează în ordine lexicografică toate tripletele vocală-consoană-vocală formate cu literele mari A,B,C,D,E: ABA,ABE, ACA, ACE, ADA, ADE, EBA, EBE, ECA, ECE, EDA, EDE. Dacă se generează, folosind aceeași metodă și aceleași litere, toate tripletele consoană-vocală-consoană, stabiliți care dintre următoarele variante este o secvență de triplete generate unul imediat după celălalt:

a) ACE ADA ADE b) BEC BED CAB c) BEC CEC DEC
d) CEA CEB CEC e) DAC DAB DEB f) DAD DAC DAB

8. Se consideră numerele naturale m și n ($0 \leq m \leq 10$, $0 \leq n \leq 10$) și subprogramul **Ack(m,n)**, care calculează valoarea funcției **Ackermann** pentru valorile m și n . Precizați numărul de apeluri recursive ale subprogramului **Ack** pentru valorile $m=1$ și $n=2$, **Ack(1,2)**.

Limbajul C++/C

```
int Ack(int m, int n)
{if(m==0)
    return n+1;
else
    if(m>0 && n==0)
        return Ack(m-1,1);
    else
        return Ack(m-1,Ack(m,n-1));
}
```

Limbajul Pascal

```
Function
Ack(m,n:integer):integer;
begin
    if m=0 then
        Ack:=n+1
    else
        if (m>0) AND (n=0) then
            Ack:=Ack(m-1,1)
        else
            Ack:=Ack(m-1,Ack(m,n-1));
end;
```

a) de 5 ori

b) de 6 ori

c) de 7 ori

d) de 8 ori

e) de 9 ori

f) de 10 ori

9. Un arbore cu rădăcină are 359 de noduri numerotate de la 1 la 359. Dacă vectorul de tați al acestui arbore (vector notat cu t) are proprietatea că $t[i] = \lfloor \frac{i}{2} \rfloor$, pentru orice i de la 1 la 359, unde $[x]$ reprezintă partea întreagă a numărului x , atunci numărul de noduri care au exact un descendent direct în acest arbore este:

a) 178

b) 9

c) 4

d) 3

e) 1

f) 0

10. Fie A un tablou unidimensional cu n elemente și subprogramul **Swap** care realizează interschimbarea valorilor pe care le primește. Atunci următoarea secvență de cod sortează descrescător tabloul A . Câte apeluri ale subprogramului **Swap** vor fi făcute dacă inițial $A[i]=i$ pentru $i=1, 2, \dots, n$?

Limbajul C++/C

```
for (j=1; j<=n-1; j++)
  for (k=1; k<=n-j; k++)
    if (A[k]<A[k+1])
      Swap(A[k], A[k+1]);
```

Limbajul Pascal

```
for j:=1 to n-1 do
  for k:=1 to n-j do
    if A[k]<A[k+1] then
      Swap(A[k], A[k+1]);
```

a) $n/2$ b) n c) $n-1$ d) $n(n-1)$ e) $n(n-1)/2$ f) $n*n$

11. Care dintre următoarele grafuri neorientate este un graf **eulerian**, dar **nu este hamiltonian**? Grafurile sunt precizate prin n numărul de noduri și mulțimea U a muchiilor.

a) $n=3, U=\{[1,2], [1,3], [2,3]\}$ b) $n=4, U=\{[1,2], [1,3], [2,3], [1,4], [2,4], [3,4]\}$ c) $n=5, U=\{[1,3], [1,4], [3,4], [2,4], [4,5], [2,5]\}$ d) $N=6, U=\{[1,2], [2,3], [3,4], [5,4], [6,5], [2,6]\}$ e) $N=6, U=\{[1,2], [2,3], [3,4], [5,4], [6,5], [1,6]\}$

f) nici unul din grafurile anterioare

12. Care este instrucțiunea echivalentă cu instrucțiunea de mai jos care să conțină o singură instrucțiune **if**, unde x , y și z sunt variabile care rețin valori întregi:

Limbajul C++/C

```
if (x>y)
  if (y>z)
    if (z>x) s=x+y+z;
    else p=x*y*z;
```

Limbajul Pascal

```
if x>y then
  if y>z then
    if z>x then s:=x+y+z;
    else p:=x*y*z;
```

Limbajul C++/C

a) `if (x>y && y>z)`b) `if (x>y && y>z)`c) `if (x>y || y>z)`

<code>s=x+y+z;</code>	<code>p=x*y*z;</code>	<code>s=x+y+z;</code>
<code>else p=x*y*z;</code>		
d) <code>if (x>y && y>z)</code>	e) <code>if (!(x>y) && y>z)</code>	f) <code>if (!(x>y && y>z))</code>
<code>s=x+y+z;</code>	<code>s=x+y+z;</code>	<code>s=x+y+z;</code>
	<code>else p=x*y*z;</code>	<code>else p=x*y*z;</code>

Limbajul Pascal

a)	b)	c)
<code>if (x>y) AND (y>z)</code>	<code>if (x>y) AND (y>z)</code>	<code>if (x>y) OR (y>z)</code>
<code>then s:=x+y+z</code>	<code>then p:=x*y*z;</code>	<code>then</code>
<code>else</code>		<code>s:=x+y+z;</code>
<code>p:=x*y*z;</code>		
d)	e)	f)
<code>if (x>y) AND (y>z)</code>	<code>if not (x>y) AND (y>z)</code>	<code>if not ((x>y) AND</code>
<code>then</code>	<code>then s:=x+y+z</code>	<code>(y>z))</code>
<code>s:=x+y+z;</code>	<code>else p:=x*y*z;</code>	<code>then s:=x+y+z</code>
		<code>else p:=x*y*z;</code>

13. Dacă se dorește căutarea unui număr într-un șir ordonat de 1000 de elemente, folosind algoritmul căutării binare, atunci numărul maxim al pașilor efectuați (comparații) va fi:
- | | | |
|-----------------------------|--------------------------|------------------------------|
| a) cuprins între 1 și 6 | b) cuprins între 7 și 12 | c) cuprins între 13 și 18 |
| d) Cuprins între 450 și 500 | e) exact 500 | f) cuprins între 999 și 1001 |

14. Precizați care dintre următoarele subprograme calculează corect *cifra de control* a unui număr natural asociat *x* (*Cifra de control* a unui număr natural se determină calculând suma cifrelor numărului, apoi suma cifrelor sumei și așa mai departe până când suma obținută reprezintă un număr cu o singură cifră).

Limbajul C++/C

A.

```
int Control(int x)
{
    int s;
    if (x>9){
        s=x%10+Control(x/10);
        if(s<10) return s;
        else
            return Control(s);
    }
    else return x;
}
```

B.

```
int Control(int x)
```

Limbajul Pascal

A. function

```
Control(x:integer): integer;

var s:integer;
begin
    if x>9 then
        begin
            s:=x mod 10+Control(x div 10);
            if s<10 then Control:=s
            else
                Control:=Control(s);
        end
    else Control:=x;
end;
```

B. function

```

{int s;                                Control(x:integer):integer;
  if (x>9){
    s=x%10+Control(x/10);
    if(s < 10)
      return s;
    else
      return Control(s);
  }
  return s;
}

```

```

var s:integer;
begin
  if x>9 then
    begin
      s:=x mod 10 +
        Control(x div 10);
      if s<10 then
        Control:=s
      else
        Control:=Control(s);
    end;
    Control:=s;
  end;
end;

```

C.

```

int Control(int x)
{int s;
  if (x >10)
    s=x%10+Control(x/10);
    return x;
}

```

```

C. function
Control(x:integer):integer;
  var s:integer;

  begin
    if x>10 then s:=x mod 10
    +Control(x div 10);
    Control:=x
  end;
end;

```

a) A
d) B,C

b) A,B
e) C

c) B
f) niciunul

15. Fie G un graf neorientat conex, cu mulțimea de vârfuri V , având proprietățile: (1) fiecare vârf are cel mult 3 vecini și (2) există un vârf $u \in V$ astfel încât pentru orice $v \in V$ avem $d(u; v) \leq 5$, unde $d(u; v)$ reprezintă lungimea celui mai scurt drum dintre vârfurile u și v (ca număr de muchii). Care este numărul maxim de vârfuri din G ?

a) 46
d) 100

b) 64
e) 150

c) 94
f) 194

Varianta 42

1. Precizați valoarea afișată în urma executării secvenței de program:

Limbajul C++/C

```
int v[8] =
{2,4,1,3,11,5,1,3};
int ex(int n, int m)
{
    if (n==m)
        return v[n]%2;
    else
        return ex(n, (n + m) /
2) + ex((n + m)/2 + 1, m);
}
int main()
{cout<< ex(0,7); |
printf("%d",ex(0,7));
return 0;}
```

a) 2 b) 3 c) 5

Limbajul Pascal

```
var v:array[0..7] of integer
=(2,4,1,3,11,5,1,3);
function
ex(n, m:integer):integer;
begin
    if n=m
    then ex:=v[n] mod 2
    else
        ex:= ex(n, (n+m) div 2) +
ex((n + m) div 2 + 1, m);
    end;
begin
    write(ex(0,7));
end.
```

d) 6 e) 7 f) 9

2. Se consideră un șir de caractere **a** care conține cel puțin o majusculă. Care dintre cele trei secvențe **S1**), **S2**) și **S3**) de mai jos afișează prima literă mare din șir?

Limbajul C++/C

```
//S1
i = 0;
while (a[i]&&(a[i]<'A' || a[i]
>'Z')) i++;
cout<< a[i];
| printf("%c",a[i]);
```

```
//S2)
i=0;
do { i++;} while (a[i] &&
!(a[i]>='A' && a[i]<= 'Z'));
cout<<a[i];
| printf("%c",a[i]);
```

```
//S3)
for(i=0;a[i]&& a[i]>='A'&&
a[i]<='Z';i++)
cout<<a[i];
| printf("%c",a[i]);
```

a) Numai S1
d) S2 și S3

Limbajul Pascal

```
//S1
i:= 1;
while (i<=length(a)) and ((a[i]
<'A') or (a[i] >'Z')) do i:=i+1;
write(a[i]);
```

```
//S2)
i:=1;
repeat i:=i+1;
until not((i<=length(a))and
((a[i]<'A') or (a[i] >'Z')));
write(a[i]);
```

```
//S3
i:= 1;
while (i<=length(a))and
((a[i]>='A') and (a[i] <='Z')) do
begin
    write(a[i]);
    i:=i+1
end;
```

b) Numai S2 c) S1 și S2
e) Numai S3 f) Nici una

3. Ce se va afișa în urma executării secvenței de program alăturate?

Limbajul C++/C

```
char s[]="UN11DOI22TREI33";
int i=0;
while(s[i]!='\0')
    if(s[i]=='1')
        {s[i+1]='+';i+=2;}
    else
        if(s[i]=='2')
            {s[i+1]='=';i+=2;}
        else
            if(s[i]=='3')
                {strcpy(s+i,s+i+1);i++;}
            else strcpy(s+i,s+i+1);
cout<<s; | printf("%s",s);
```

Limbajul Pascal

```
var s:string;i:integer;
begin
s:= 'UN11DOI22TREI33' ;
i:=1 ;
while i<=length(s) do
if s[i]='1' then begin
    s[i+1]:= '+' ;

    i:=i+2
end
else if s[i]='2' then begin
    s[i+1]:= '=' ;
    i:=i+2
end
else if s[i]='3' then begin

    delete(s,i,1);
    i:=i+1
end
else delete(s,i,1);
write(s);
end.
```

a) UN+DOI=TREI

b) UN+DOI=3

c) 1+2=TREI

d) 1+2=3

e) 11+22=33

f) 111+222=333

4. Variabilele întregi i și x memorează numere naturale. Tabloul unidimensional v conține numerele naturale 0,1,2,0,4,5,6. Numărul de elemente cu valoarea 10 din v , după rularea secvenței de mai jos este?

Limbajul C++/C

```
i=0; x=10;
do
{
    v[i++]=x;
} while (i < 6 &&
v[i]);
```

a) zero

d) trei

Limbajul Pascal

```
i:=0; x:=10;
repeat
    v[i]:=x; i:=i+1;
until not ((i < 6)and(v[i]<>0));
```

b) unul

e) cinci

c) doua

f) șapte

5. Se consideră următoarea secvență. Stabiliți ce se afișează în urma rulării?

Limbajul C++/C

```
#include <iostream>
using namespace std;
```

Limbajul Pascal

```
var v:array[1..100] of
integer;
```

```

int main()
{
int l,lmax,umax,i,n,v[100];
cin>>n;
| scanf("%d",&n);
for(i=1;i<=n;i++)
cin>>v[i];
| scanf("%d",&v[i]);
l=lmax=1;
for(i=2;i<=n;i++)
if(v[i]%2+v[i-1]%2==1) l++;
else {
if(l>lmax)
{lmax=l;
umax=i-1;}
l=1;
}
if(l>lmax)
{lmax=l;umax=n;}
for(i=umax-lmax+1;i<=umax;i++)
cout<<v[i]<<' ';
| printf("%d",v[i]);
return 0;
}

```

```

l,lmax,umax,i,n:integer;
begin
readln(n);
for i:=1 to n do
readln(v[i]);
l:=1; lmax:=1;
for i:=2 to n do
if v[i] mod 2+v[i-1] mod
2=1
then
l:=l+1
else
begin
if l>lmax then
begin
lmax:=l;
umax:=i-1
end;
l:=1
end;
if l>lmax then begin
lmax:=l; umax:=n
end;
for i:=umax-lmax+1 to umax
do write(v[i],' '); end.

```

- | | | |
|---|--|---|
| a) Cea mai lungă
secvență de valori
de aceeași paritate | b) Cea mai scurtă
secvență de valori
distincte | c) Cea mai lungă
secvență de valori
egale |
| d) Cea mai lungă
secvență de valori
impare | e) Cea mai lungă
secvență de valori
pare | f) Cea mai lungă
secvență de valori
de parități
diferite |

6. Se consideră un graf neorientat cu 10 vârfuri numerotate de la 1 la 10, graf cu proprietatea că există muchie între vârfurile i și j dacă și numai dacă i și j sunt numere prime între ele. Care este suma gradelor vârfurilor acestui graf?

- | | | |
|-------|-------|-------|
| a) 20 | b) 30 | c) 32 |
| d) 50 | e) 60 | f) 62 |

7. Numărul maxim de muchii într-un graf neorientat cu n vârfuri și p componente conexe ($p < n$) este:

- | | | |
|----------------|----------------|--------------------------------------|
| a) $n \cdot p$ | b) $n - p + 1$ | c) $((n - p) \cdot (n - p - 1)) / 2$ |
|----------------|----------------|--------------------------------------|

d) C^2_{n-p} e) C^2_{n-p+1} f) C^2_n

8. Funcția recursivă **suma** trebuie definită astfel încât apelul **suma(n)** să returneze suma pătratelor perfecte mai mici sau egale cu **n**. Care este expresia cu care trebuie completată definiția funcției?

Limbajul C++/C

```
long suma (int i)
{
    if(i==0) return 0;
    int j=sqrt(i);
    return ...;
}
```

Limbajul Pascal

```
function
suma (i:longint):longint;
var j:longint;
begin
    if i = 0 then suma:=0
    else begin
        j:=trunc(sqrt(i));
        suma:= ...
    end end;
```

- a) $j*j+suma(j-1)$ b) $j*j+suma(j)$ c) $j*j+suma(j*j-1)$
d) $j+suma(j*j-1)$ e) $j+suma(j-1)$ f) $j*j+suma(j*j)$

9. Valoarea variabilei **n** la finalul rulării următorului program este?

Limbajul C++/C

```
int main()
{
    int v[]={0, 1, 2, 3, 4, 5, 0};
    int i=0, n=0;
    do{ if (i==v[i]) n++;
        } while(i<6 && v[i++]);
    return 0;
}
```

Limbajul Pascal

```
var v:array[0..6] of
integer=(0, 1, 2, 3, 4, 5,
0); i,n:integer;
begin
    i:=-1; n:=0;
    repeat
        i:=i+1;
        if i=v[i] then
            n:=n+1;
        until not((i<6)and
(v[i]<>0))
    end.
```

- a) 0 b) 1 c) 3
d) 5 e) 7 f) programul va intra într-un ciclu infinit

10. Un arbore binar complet este un arbore cu rădăcină în care fiecare nod are exact doi fii sau niciunul. Știind că rădăcina se găsește pe nivelul 0, numărul maxim de noduri de pe nivelul 5 dintr-un astfel de arbore este?

- a) 15 b) 2^4 c) 31
d) 2^5 e) 50 f) 120

11. Un elev, folosind metoda backtracking, construiește toate numerele cu cifre distincte, numere care au suma cifrelor egală cu 5 și nu sunt divizibile cu 10. El obține, în această ordine, numerele: 104; 14; 203; 23; 302; 32; 401; 41; 5. Folosind aceeași metodă, el construiește toate numerele naturale cu cifre diferite, nedivizibile cu 10 și cu suma cifrelor egală cu 6. Care sunt primele patru numere pe care le construiește?
- a) 114;123;132;141 b) 123;132;15;213 c) 1023;123;1032;132
d) 1023;1032;105;1203 e) 1014;105;15;6 f) 1023;105;15;6
12. Căutarea unui element într-un vector sortat descrescător se realizează în mod eficient cu un algoritm care utilizează:
- a) metoda căutării binare b) metoda backtracking c) sortarea crescătoare a vectorului
d) parcurgerea elementelor de pe poziții pare e) metoda căutării secvențiale f) parcurgerea elementelor de pe poziții impare
13. Generarea tablourilor bidimensionale pătratice de ordinul n , cu elemente 0 și 1, cu proprietatea că pe fiecare linie și pe fiecare coloană există un singur element egal cu 1, se poate realiza utilizând metoda backtracking. Algoritmul utilizat este echivalent cu algoritmul de generare a:
- a) produsului cartezian b) combinărilor c) aranjamentelor
d) permutărilor e) submulțimilor f) partițiilor
14. Fie graful orientat $G=(V,E)$ unde $\text{card}(V)=20$ iar $E=\{(i,j) \mid i < j, i,j \in V\}$. Numărul de componente tare conexe ale grafului G este:
- a) 1 b) 5 c) 8
d) 10 e) 15 f) 20
15. Stabiliți valoarea apelului $F(11111)$.
- Limbajul C++/C**

```
int F(int N)
{ if (N==1) return 0;
  return 1+F(N/2);
}
```

Limbajul Pascal

```
function
F(N:integer):integer;
begin
  if N=1 then F:=0
  else
    F:=1+F(N div 2);
  end;
```
- a) 8 b) 10 c) 13
d) 20 e) 36 f) 50

Varianta 43

1. Variabila **v** reține un număr întreg. Indicați ce valoare va avea **v** după executarea următoarei secvențe de instrucțiuni?

Limbajul C++/C

```
int v=2;  
v=v*v; v=2*v+v/2;  
v=v-4;  
v=v%5;
```

a) 0 b) 1 c) 2

Limbajul Pascal

```
v:=2;  
v:=v*v;  
v:=2*v+v div 2; v:=v-4;  
v:=v mod 5;
```

d) 8 e) 10 f) 16

2. Variabilele **x** și **y** sunt reale, iar **i** și **j** sunt întregi. Indicați ce valoare va avea **x** după executarea următoarei secvențe de instrucțiuni?

Limbajul C++/C

```
x=1.5; y=2.1;  
i=x; j=y+1;  
x=x*y+i/j;
```

a)2 b) 2.15 c) 3

Limbajul Pascal

```
x:=1.5; y:=2.1;  
i:=trunc(x);  
j:=trunc(y)+1;  
x:=x*y+i div j;
```

d) 3.1 e) 3.15 f) 3.5

3. Variabilele **i**, **j** și **k** sunt de tip întreg. Indicați ce valoare se afișează după executarea următoarei secvențe?

Limbajul C++/C

```
i=0; j=0;  
if(j) j--; else i++;  
if(i) i--; else j++;  
k=i+j;  
cout<<k; | printf("%d",k);
```

a) -3 b) -2 c) -1

Limbajul Pascal

```
i:=0; j:=0;  
if j<>0 then dec(j)  
          else inc(i);  
if i<>0 then dec(i)  
          else inc(j);  
k:=i+j;  
write(k);
```

d) 0 e) 1 f) 2

4. Variabilele **n**, **p** și **x** sunt de tip întreg. Indicați ce valoare se afișează după executarea următoarei secvențe?

Limbajul C++/C

```
n=25198764, p=1, x=0;  
while(n)  
{if(n%2)  
  {p=p*10;  
  x=x+n%10*p;  
 }  
  n/=10;  
}
```

Limbajul Pascal

```
n:=25198764; p:=1; x:=0;  
while n<>0 do  
  begin  
    if n mod 2<>0 then  
      begin  
        p:=p*10;  
        x:=x+n mod 10 * p  
      end;  
  end;
```

```
cout<<x; |printf("%d",x);
```

```
n:=n div 10
end;
write(x);
```

a) 2864 b) 4682 c) 5197 d) 7915 e) 51970 f) 79150

5. În secvența de mai jos toate variabilele sunt de tip întreg. Indicați ce valoare va fi afișată pentru **n=52**?

Limbajul C++/C

```
cin>>n; |scanf("%d",&n);
x=2;
for(i=3; i<=n;i++)
{
  x*=i;
  while(!(x%10)) x/=10;
  x%=10;
}
cout<<x; |printf("%d",x);
```

Limbajul Pascal

```
readln(n); x:=2;
for i:=3 to n do
begin
  x:=x*i;
  while x mod 10=0 do
    x:=x div 10;
  x:=x mod 10
end;
write(x);
```

a) 2 b) 4 c) 5 d) 6 e) 7 f) 8

6. Indicați ce valoare se va afișa după executarea secvenței următoare?

Limbajul C++/C

```
int v[15]={28,0,56,4,0,0,
13,6,0,18,0,26,90,0,25};
int k=15, i,j;
for(i=0;i<k;i++)
if(!v[i])
{
  for(j=i;j<k-1;j++)
    v[j]=v[j+1];
  v[j]=0; k--; i--;
}
for(i=0;i<k;i++)
cout<<v[i]<<" ";
|printf("%d",v[i]);
```

Limbajul Pascal

```
var v:array[0..14] of
integer =
(28,0,56,4,0,0,13,6,0,18,0,
26,90,0,25);
k,i,j:integer;
i:=0; k:=15;
while i<k do
begin
  if v[i]=0 then
  begin
    for j:=i to k-2 do
      v[j]:=v[j+1];
    v[j+1]:=0;
    dec(k);
    dec(i)
  end;
  inc(i)
end;
for i:=0 to k-1 do
write(v[i], ' ');
```

a) 4 6 13 18 25 26 28 56 90
b) 4 6 13 18 25 26 28 56 90 0 0 0
c) 4 6 13 18 25 26 28 56 90 0 0 0 0 0
d) 28 56 4 13 6 18 26 90 25
e) 28 56 4 13 6 18 26 90 25 0 0 0
f) 28 56 4 13 6 18 26 90 25 0 0 0 0 0

7. Indicați ce se va afișa după executarea secvenței de program?

Limbajul C++/C

```
int main()
{
char s[100]="Poli-
informatica", *p;
p=strchr(s, '-');
(p+1)[0]-=32;
s[p-s]='\0';
s[0]+=32;
strcat(p+1, "-");
strcat(p+1, s);
strcpy(s, p+1);
cout<<s; | printf("%s", s);
}
```

Limbajul Pascal

```
var s,p:string; i:integer;
begin
s:='Poli-informatica';
i:=pos('-',s);
s[i+1]:=chr(ord(s[i+1])-32);
s[1]:=chr(ord(s[1])+32);
p:=copy(s,i+1,length(s)-i);
delete(s,i,length(s)-i+1);
s:=p+'-'+s;
write(s);
end.
```

- a) INFO-POLI b) Informatica-poli c) informatica
d) info-poli e) informatica-poli f) poli
8. Indicați ce se va afișa după executarea următoarei secvențe de program?

Limbajul C++/C

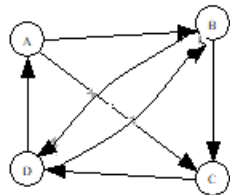
```
typedef struct
{char *p;}S;
int main()
{char p[10]="abcd";
S S[2];
int i;
for(i=0;i<2;i++)
S[i].p=p+i;
cout<<S[1].p[0];
| printf("%c",S[1].p[0]);
return 0;
}
```

Limbajul Pascal

```
type ST=record p:string
end;
var p:string;
S:array[0..1] of ST;
i:integer;
begin
p:='abcd';
for i:=0 to 1 do
begin
S[i].p:=copy(p,i+1,
length(p)-i+1);
end;
write(S[1].p[1]);
end.
```

- a) a b) b c) c d) d e) e f) f

9. Pentru graful orientat alăturat, indicați între care perechi de noduri există drum



de lungime 5?

- a) A și B b) A și C c) A și D d) B și C e) B și D f) C și A

10. Fie funcția recursivă de mai jos. Ce valoare va avea apelul $f(f(f(f(0))))$?

Limbajul C++/C

```
int f(int x)
{if(x<7) return f(x+2)+1;
  else return x-5;}
```

Limbajul Pascal

```
function
f(x:integer):integer;
begin
if x<7 then f:=f(x+2)+1
  else f:=x-5;
end;
```

- a) 1 b) 2 c) 3 d) 4 e) 5 f) 6

11. Indicați ce se va afișa în urma executării următoarei secvențe?

Limbajul C++/C

```
int a[6][6]={ {2,4,1,5,3},
{5,1,4,2,3}, {1,2,3,4,5}, {5,4,3,2,1}, {4,1,5,3,2} };
int i,j,k=3;
for(int m=0;m<k;++m)
{for(i=4;i>1;--i)
  for(j=0;j<5;++j)
    a[i+1][j]=a[i][j];
  for(j=0;j<5;j++)
    a[2][j]=a[5][j];
}
for(i=0;i<5;++i)
{for(j=0;j<5;j++)
  cout<<a[i][j]<<" ";
  |printf("%d ",a[i][j]);
  cout<<endl;|printf("\n");
}
```

Limbajul Pascal

```
var a:array[0..5,0..4]of
integer=((2,4,1,5,3),(5,1,4,2,3),(1,2,3,4,5),(5,4,3,2,1),
(4,1,5,3,2),(0,0,0,0,0));
i,j,k,m:integer;
k:=3;
for m:=0 to k-1 do
begin
  for i:=4 downto 2 do
    for j:=0 to 4 do
      a[i+1,j]:=a[i,j];
    for j:=0 to 4 do
      a[2,j]:=a[5,j]
    end;
  for i:=0 to 4 do
    begin
      for j:= 0 to 4 do
        write(a[i,j], ' ');
      writeln
    end;
```

- | | | | | | |
|-----------|-----------|-----------|-----------|-----------|-----------|
| a) | b) | c) | d) | e) | f) |
| 2 3 1 5 4 | 2 4 1 5 3 | 2 4 1 5 3 | 2 4 1 5 3 | 2 4 1 5 3 | 3 4 1 5 2 |
| 5 1 4 2 3 | 4 1 5 3 2 | 5 1 4 2 3 | 5 4 3 2 1 | 5 4 3 2 1 | 4 1 5 3 2 |
| 1 2 3 4 5 | 5 4 3 2 1 | 1 2 3 4 5 | 4 1 5 3 2 | 4 1 5 3 2 | 5 4 3 2 1 |
| 5 4 3 2 1 | 1 2 3 4 5 | 5 4 3 2 1 | 5 1 4 2 3 | 5 1 4 2 3 | 1 2 3 4 5 |
| 2 1 5 3 4 | 5 1 4 2 3 | 4 1 5 3 2 | 1 2 3 4 5 | 1 2 3 5 4 | 3 1 4 2 5 |

12. Folosind metoda backtracking, se determină în ordine crescătoare, toate numerele de 4 cifre distincte, oricare două cifre alăturate neputând fi prime. Primele 7 soluții sunt: **1024, 1026, 1028, 1029, 1034, 1036, 1038, ...** Indicați care sunt cele 2 numere generate înaintea soluției **7401**?

a) 7091 b) 7195 c) 7196 d) 7297 e) 7397 f) 7916
 7092 7198 7198 7298 7398 7918

13. Secvența gradelor dintr-un graf neorientat este formată din gradele tuturor nodurilor grafului, aranjate în ordine descrescătoare. Indicați care dintre următoarele secvențe nu poate fi secvență a gradelor pentru niciun graf?

I: 7, 6, 5, 4, 4, 3, 2, 1

II: 6, 6, 6, 6, 3, 3, 2, 2

III: 7, 6, 6, 4, 4, 3, 2, 2

IV: 8, 7, 7, 6, 4, 2, 1, 1

a) I și II b) I și IV c) II și IV d) III și IV e) doar II f) doar IV

14. Fie următoarele funcții recursive de mai jos. Indicați care este complexitatea timp a celor două funcții?

Limbajul C++/C

```
int f1(int n)
{ if(n<=1) return 0;
  return 2*f1(n-1);
}
int f2(int n)
{ if(n<=1) return n;
  return f2(n-1)+f2(n-1);
}
```

a) $O(n^2)$ pentru amândouă
 c) $O(2^n)$ pentru f1 și $O(n)$ pentru f2
 e) $O(3^n)$ pentru f1

Limbajul Pascal

```
function
f1(n:integer):integer;
begin
  if n<=1 then
    f1:=0; f1:=2*f1(n-1)
  end;
function
f2(n:integer):integer;
begin
  if n<=1 then f2:=n;
  f2:=f2(n-1)+f2(n-1)
end;
```

b) $O(n)$ pentru f1 și $O(2^n)$ pentru f2
 d) $O(n)$ pentru amândouă
 f) $O(3^n)$ pentru f1 și f2

15. Indicați care este complexitatea următoarei secvențe de instrucțiuni:

Limbajul C++/C

```
int n,k,p;
cin>>n>>k; | scanf("%d%d", &n, &k);
p=1;
while(k>0)
```

Limbajul Pascal

```
var n,k,p:integer;
readln(n,k);
p:=1;
while k>0 do
```

```
if(k%2) p*=n, k--;  
else n*=n, k/=2;
```

```
if k mod 2<>0 then  
  begin p:=p*n; dec(k)  
  end  
else  
  begin n:=n*n; k:=k div 2  
  end;
```

a) $O(\log_2 k)$ b) $O(\log_2 n)$ c) $O(k)$

d) $O(n)$ e) $O(n^2)$ f) $O(k^2)$

Varianta 44

1. Variabila **v** reține un număr întreg. Indicați ce valoare va avea **v** după executarea următoarei secvențe de instrucțiuni?

Limbajul C++/C

```
v=2;
v=5*v-v%3;
v=v+v/3;
```

a) 7 b) 8 c) 10

Limbajul Pascal

```
v:=2;
v:=5*v-v mod 3;
v:=v+v div 3;
```

d) 11 e) 12 f) 16

2. Variabilele **x** și **y** sunt reale, iar **i** și **j** sunt întregi. Indicați ce valoare va avea **x** după executarea următoarei secvențe de instrucțiuni?

Limbajul C++/C

```
x=1.5; y=2.0;
i=2; j=4;
x=x*y+(float)i/j;
```

a) 2.15 b) 2.5 c) 3.0

Limbajul Pascal

```
x:=1.5; y:=2.0;
i:=2; j:=4;
x:=x*y+i/j;
```

d) 3.15 e) 3.5 f) 4.0

3. Variabilele **i** și **j** sunt întregi. Indicați ce se va afișa după executarea următoarei secvențe?

Limbajul C++/C

```
i=2; j=3;
if(j) j--;
    else if(i) i++;
        else j++;
if(!j) i--;
    else if(i) j++;
        else j=0;
cout<<i+j; |printf("%d",i+
j);
```

a) 0 b) 1 c) 2

Limbajul Pascal

```
i:=2; j:=3;
if j<>0 then dec(j)
    else if i<>0 then
        inc(i)
            else inc(j);
if not(j<>0) then inc(i)
    else if(i<>0) then
        inc(j)
            else j:=0;
write(i+j);
```

d) 3 e) 4 f) 5

4. Indicați ce se va afișa după executarea următoarei secvențe de program?

Limbajul C++/C

```
int main()
{
char s[11]="ABCDE",aux[11];
strcat(s+2,"ABCDE");
strcpy(aux, s+3);
strcpy(s,aux);
cout<<s[0]-s[2];
|printf("%d",s[0]-s[2]);
return 0;}
```

Limbajul Pascal

```
var s:string;
begin
s:='ABCDE';
s:=s+'ABCDE';
delete(s,1,3);
write(ord(s[1])-ord(s[3]));
end.
```

- a) 0 b) 1 c) 2 d) 3 e) 4 f) 5

5. Indicați ce se va afișa după executarea următoarei secvențe de program?

Limbajul C++

```
typedef struct
    { int S;}S;
int f(S &s)
{return -- s.S;}
int main()
{ int i;
  S S={2};
  i=f(S);
  cout<<i;
  return 0;
```

Limbajul C

```
typedef struct
{ int S;}S;
int f(S *s)
{
  return --(*s).S;}
int main()
{ int i;
  S S={2};
  i=f(&S);
  printf("%d",i);
  return 0;
}
```

Limbajul Pascal

```
type ST=record
  S:integer end;
var S:ST=(S:2);
      i:integer;
function f(var
  s:ST):integer;
begin
  dec(s.S);f:=s.S
end;
begin
  i:=f(S);write(i)
end.
```

- a) 0 b) 1 c) 2 d) 3 e) 4 f) eroare de compilare

6. Indicați cu ce instrucțiune trebuie înlocuite punctele de suspensie din următoarea secvență, astfel încât aceasta să afișeze numărul de divizori pozitivi ai lui **n** (număr natural nenul)?

Limbajul C++/C

```
int p=1,d,e;
for (d=2;d*d<=n;++d)
{
    for (e=1;n%d==0;e++)
        n/=d;
        p*=e;
}
if (n>1) .....
cout<<p; | printf("%d",p);
```

Limbajul Pascal

```
var p,d,e:integer;
p:=1; d:=2;
while d*d<=n do
begin
    e:=1;
    while n mod d=0 do
begin
        n:=n div d;
        inc(e)
    end;
    p:=p*e
end;
if n>1 then ...
write(p);
```

a)
Limbajul C++/C
 p*=2;
Limbajul Pascal
 p:=p*2;
 d)Limbajul C++/C

b)
Limbajul C++/C
 p*=d;
Limbajul Pascal
 p:=p*d;
 e)Limbajul C++/C

c)
Limbajul C++/C
 p*=d+1;
Limbajul Pascal p:=p*(d+1);
 f)Limbajul C++/C

p*=e;
Limbajul Pascal
 p:=p*e;

p*=(e+1);
Limbajul Pascal
 p:=p*(e+1);

p*=(d+1);
Limbajul Pascal p:=p*(d+1);

7. Următoarea secvență afișează numărul de perechi de elemente din vectorul **v**, cu proprietatea că suma celor două elemente din pereche este divizibilă cu **k** (**k<100**). Se numără numai perechile (**v[i];v[j]**) cu proprietatea enunțată care au **i<j**, unde **i** și **j** sunt numere naturale, **i<n**, **j<n**. Indicați cu ce expresie trebuie înlocuite punctele de suspensie?

Limbajul C++/C
 int v[100],n,k;
 cin>>n>>k;
 |scanf("%d%d",&n,&k);
 for(int i = 0; i<n; ++i)

 cin>>v[i]; |scanf("%d",&v
 [i]);
 int x[100]={0};
 for (int i=0; i<n;i++)
 ++x[v[i]%k];
 int sum = x[0]*(x[0]-
 1)/2;
 for(int i=1; i<=k/2 &&
 i!=(k-i); i++)
 sum += x[i] * x[k-i];
 if (k % 2 == 0) sum +=...;
 cout<<sum;
 |printf("%d",sum);

Limbajul Pascal
 var n,k,i,sum:integer;
 v,x:array[0..99] of integer;
 begin
 readln(n,k);
 for i:=0 to n-1 do
 readln(v[i]);
 for i:=0 to n-1 do
 inc(x[v[i] mod k]);
 sum:=x[0]*(x[0]-1)div 2;
 i:=1;
 while (i<=k div 2) and
 (i<>(k-i)) do
 begin
 sum:=sum+x[i]*x[k-i];
 inc(i)
 end;
 if k mod 2=0 then
 sum:=sum+...;
 write(sum);
 end.

a) x[k]

b) x[n]

c) x[k]*x[k]

d)

Limbajul C++/C x[k]*(x[k]-1)/2

e)

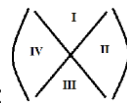
Limbajul C++/C x[k/2]*(x[k/2]-1)/2

Limbajul Pascal x[k]*(x[k]-1) div 2

f) x[n]*x[n]

Limbajul Pascal x[k div 2]*(x[k div 2]-1) div
 2

8. Variabilele i și j sunt de tip întreg, iar variabila a memorează un tablou bidimensional cu n linii și n coloane, numerotate de la 0 la $n-1$, având inițial toate elementele egale



cu -1 . Matricea se împarte în 4 cadrane astfel:

Indicați cu ce instrucțiune trebuie înlocuite punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul a să memoreze în cadranul I, doar valoarea 1 , în cadranul II, doar valoarea 2 , în cadranul III doar valoarea 3 iar în ultimul cadran doar valoarea 4 .

Limbaj C++/C

```
for (i=0; i<n/2; i++)
    for (j=i+1; j<n-i-1;
        j++)
    {
        a[i][j]=1;
        ...;
        a[n-i-1][j]=3;
        a[j][i]=4;
    }
```

a)

Limbajul C++/C $a[i][n-j]=2;$

Limbajul Pascal $a[i,n-j]:=2;$

c)

Limbajul C++/C $a[n-j-1][n-i-1]=2;$

Limbajul Pascal $a[n-j-1,n-i-1]:=2;$

e)

Limbajul C++/C $a[j][n-i]=2;$

Limbajul Pascal $a[j][n-i]:=2;$

Limbaj Pascal

```
for i:=0 to n div 2-1 do
    for j:=i+1 to n-i-2 do
        begin
            a[i,j]:=1;
            ...;
            a[n-i-1,j]:=3;
            a[j,i]:=4
        end;
```

b)

Limbajul C++/C $a[n-i][n-j]=2;$

Limbajul Pascal $a[n-i,n-j]:=2;$

d)

Limbajul C++/C $a[n-j][n-i]=2;$

Limbajul Pascal $a[n-j,n-i]:=2;$

f)

Limbajul C++/C $a[i][i]=2;$

Limbajul Pascal $a[i][i]:=2;$

9. Fie funcția recursivă de mai jos. Indicați ce valoare va avea apelul $f(16)$?

Limbajul C++/C

```
int f(int x)
{ if(x>8)
    return f(f(x-3))+4;
  else return x-5;
}
```

a) -5

b) -2

c) -1

Limbajul Pascal

```
function f(x:integer):integer;
begin
    if x>8 then
        f:=f(f(x-3))+4
    else f:=x-5;
end;
```

d) 1

e) 2

f) 5

10. Indicați câte componente tare conexe are graful orientat $G=(V,E)$ unde $V=\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ iar $E=\{(1,2), (1,7), (1,10), (2,6), (3,2), (3,5), (3,9), (4,3), (4,6), (5,2), (6,1), (8,7), (9,6), (9,8)\}$?

a) 6

b) 7

c) 8

d) 9

e) 10

f) 5

11. Folosind metoda backtracking, se determină în ordine lexicografică, toate cuvintele de 6 litere distincte din mulțimea $\{a, e, i, o, u, b, c, d, m, n, p\}$, oricare două litere alăturate neputând fi vocale. Primele 5 soluții sunt: **abecid**, **abecim**, **abecin**, **abecip**, **abecod**. Indicați care cuvânt este generat înaintea soluției **ebacid**?
- a) mnpdb b) apcdmn c) apmncd d) apmndc e) apnmdc f) ebdcpa
12. Fie un graf neorientat **G** cu **1002** noduri numerotate cu numere naturale consecutive de la **1** la **1002**. Știind că oricare două noduri de aceeași paritate sunt adiacente, se cere să indicați cum trebuie modificat graful, astfel încât acesta să devină eulerian.
- a) se elimină două muchii b) se elimină o muchie și se adaugă două muchii noi c) se adaugă două muchii noi
d) se elimină o muchie e) se adaugă trei muchii noi f) se elimină trei muchii
13. Fie un arbore cu rădăcină care are **5000** de noduri iar fiecare nod are maxim **4** fii. Indicați care este înălțimea minimă a arborelui?
- a) 4 b) 5 c) 6 d) 7 e) 8 f) 9
14. Indicați care este complexitatea următoarei funcții?

Limbajul C++/C

```
void f(int n)
{
    int i, j, nr=0;
    for(i=n/2; i<=n; i++)
        for(j=n; j>=1; j/=2)
            nr++;
}
```

Limbajul Pascal

```
procedure f(n:integer);
var i,j,nr:integer;
begin
    for i:=n div 2 to n do
        begin
            j:=n;
            while j>=1 do
                begin
                    inc(nr);
                    j:=j div 2;
                end
            end
        end
end;
```

- a) $O(\log n)$ b) $O(n^2)$ c) $O(n^2 \log n)$ d) $O(n \log n)$ e) $O(n)$ f) $O(n^3)$
15. Fie $x, a_0, a_1, \dots, a_n$ numere reale nenule. Indicați numărul minim de înmulțiri care este necesar pentru a calcula optim $a_0 + a_1 * x + a_2 * x^2 + a_3 * x^3 + \dots + a_n * x^n$?
- a) $n-1$ b) n c) $\frac{n}{2}$ d) $\frac{(n+1)(n+2)}{2}$ e) $n-2$ f) $\frac{(n-1)(n+1)}{2}$

Varianta 45

1. Variabilele întregi **i**, **j**, **k** memorează numere naturale. Valoarea variabilei **k** după rularea următoarei secvențe de instrucțiuni este?

Limbajul C/C++			Limbajul Pascal		
i=4; j=5; k=--i*j++;			i:=4; j:=5; dec(i); k:=i*j; inc(j);		
a) 12	b) 13	c) 14	d) 15	e) 16	f) 17

2. Variabilele întregi **i**, **j**, **k** memorează numere naturale. Valoarea variabilei **k**, după rularea următoarei secvențe de instrucțiuni, este?

Limbajul C/C++			Limbajul Pascal		
i=3; j=-3; k=i*j; k+=j; k/=i;			i:=3; j:=-3; k:=i*j; k:=k+j; k:=k div i;		
a) -8	b) -6	c) -4	d) 4	e) 6	f) 8

3. Variabilele întregi **i**, **j**, **k** memorează numere naturale. După rularea următoarei secvențe de instrucțiuni se va afișa?

Limbajul C/C++			Limbajul Pascal		
i=2; j=-2; if(j) i--; if(i) j++; k=i*j; cout<<k; printf("%d",k);			i:=2; j:=-2; if j<>0 then dec(i); if (i<>0) then inc(j); k:=i*j; write(k);		
a) -2	b) -1	c) 0	d) 1	e) 2	f) 3

4. După rularea următorului program se afișează?

Limbajul C++	Limbajul C	Limbajul Pascal
#include<iostream> using namespace std; int f(int&i) { return i++; } int main(){ int i=1,j; j=f(i); cout<<i<<' '<<j; return 0; }	#include <stdio.h> int f(int *i) { return (*i)++; } int main() { int i=1,j; j=f(&i); printf("%d %d",i,j); return 0; }	var i,j:integer; function f(var i:integer):integer; begin f:= i; inc(i) end; begin i:=1; j:=f(i); write(i, ' ',j) end.
a) 0 2	b) 1 1	c) 1 2
d) 2 1	e) 2 2	f) eroare de compilare

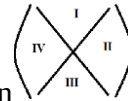
5. După rularea programului de mai jos se afișează?

Limbajul C++	Limbajul C	Limbajul Pascal
<pre>#include<iostream> using namespace std; struct S {int a[2];}; int main() { S S[2]; int i; for(i=0;i<2;i++) S[i].a[1-i]=4*!i; cout<<S[0].a[1]; return 0; }</pre>	<pre>#include <stdio.h> struct S {int a[2];}; int main() { struct S S[2]; int i; for(i=0;i<2;i++) S[i].a[1-i]=4*!i; printf("%d", S[0].a[1]); return 0;}</pre>	<pre>type ST=record a:array[0..1] of integer end; var S:array[0..1] of ST; i:integer; begin for i:=0 to 1 do S[i].a[1-i]:= 4*(1-i); write(S[0].a[1]) end.</pre>
a) 0 b) 1 c) 2	d) 3	e) 4 f) 5

6. Variabilele întregi **i**, **x** memorează numere naturale. Numărul de numere afișate după executarea următoarei secvențe de instrucțiuni este?

Limbajul C/C++	Limbajul Pascal
<pre>for(i=1;i<=10000;i++) { x=i-4; while(x>2 &&(x%10==0 x%10==2)) x=x/10, x=x-4; if(x==0 x==2) cout<<i; printf("%d",i); }</pre>	<pre>for i:=1 to 10000 do begin x:=i-4; while (x>2) and ((x mod 10=0)or(x mod 10=2)) do begin x:=x div 10; x:=x-4 end; if (x=0) or (x=2) then write(i) end;</pre>
a) 100 b) 50 c) 45	d) 40 e) 35 f) 30

7. Tabloul bidimensional **a**, pătratic, are liniile și coloanele numerotate de la 1 la 1000 și



este împărțit în 4 zone delimitate de diagonale, ca în desen. Variabilele întregi **i**, **j**, **s** memorează numere naturale. Elementele care sunt adunate în algoritmul următor se găsesc în zona/zonile?

Limbajul C/C++	Limbajul Pascal
<pre>s=0; for(i=120;i<=380;++i) for(j=i+1;j<=1000-i;j++) s+=a[1001-j][1001-i];</pre>	<pre>s:=0; for i:=120 to 380 do for j:=i+1 to 1000-i do s:=s+a[1001-j,1001-i];</pre>

- a) I b) I și II c) II d) II și III e) II și IV d) IV

8. După rularea următoarei secvențe de instrucțiuni se afișează?

Limbajul C/C++	Limbajul Pascal
<pre>char s[100]="UPB- automatica",*p; p=strchr(s,'-'); (p+1)[0]-=32; s[p-s]='\0'; p++; strcat(p,"-"); strcat(p,s); strcpy(s,p); cout<<s; printf("%s",s);</pre>	<pre>var s,p:string; i:integer; s:='UPB-automatica'; i:=pos('-',s); s[i+1]:=chr(ord(s[i+1])-32); p:=copy(s,i+1,length(s)-i); delete(s,i,length(s)-i+1); s:=p+'-'+s; write(s);</pre>
a) Auto b) Automatica-UPB c) Auto-UPB	
d) UPB e) automatica-UPB f) automatica-UPB-automatica	

9. Pentru ca tabloul unidimensional **a** să fie ordonat crescător după rularea secvenței de instrucțiuni de mai jos, punctele de suspensie se înlocuiesc cu?

Limbajul C/C++	Limbajul Pascal
<pre>int a[10]={8, 2, 1, 9, 10, 3, 7, 5, 4, 6}, b[10]={ 0 }, c[10],i, j; for(i=0;i<10;++i) for(j=i+1;j<10;++j) if(...) b[i]++; else b[j]++; for(i=0;i<10;i++) c[b[i]]=a[i]; for(i=0;i<10;i++) a[i]=c[i];</pre>	<pre>var a:array[0..9] of integer= (8,2,1,9,10,3,7,5,4,6); b, c:array[0..9] of integer; i, j:integer; for i:=0 to 9 do for j:=i+1 to 9 do if then inc(b[i]) else inc(b[j]); for i:=0 to 9 do c[b[i]]:=a[i]; for i:=0 to 9 do a[i]:=c[i];</pre>
a) $a[i] < a[j]$	b) $a[i] \leq a[j]$
c) $a[i] \neq a[j] \mid a[i] < a[j]$	d) $a[i] = a[j] \mid a[i] = a[j]$
e) $a[i] + a[j]$	f) $a[i] > a[j]$

10. Apelul **f(19,7)** are valoarea?

Limbajul C/C++	Limbajul Pascal
<pre>int f(int x, int y) { if(x>y)</pre>	<pre>function f(x,y:integer):integer; begin if x>y then f:=f(x-3,y+1)-2 else if x=y then</pre>

<pre> return f(x-3, y+1)- 2; else if(x==y) return f(x+1, y); else return 3*x- 2*y;} </pre>	<pre> f:=f(x+1,y) else f:=3*x-2*y end; </pre>
--	---

- a) -6 b) -5 c) -4 d) 4 e) 5 f) 6

11. Folosind metoda backtracking, se determină în ordine descrescătoare, toate numerele de 4 cifre nenule distincte, cifrele impare apar în ordine descrescătoare, cele pare în ordine crescătoare și oricare două cifre alăturate nu pot fi pare,. Primele 7 soluții sunt: **9875, 9873, 9871, 9853, 9851, 9831**. Cele 3 numere generate înaintea soluției **3416** sunt?

- | | |
|--------------------------|--------------------------|
| a) 3618 3421 3418 | b) 3816 4312 4316 |
| c) 4316 3618 3418 | d) 4318 4316 3618 |
| e) 4516 4318 3418 | f) 4512 4516 4518 |

12. Fie **T** un arbore oarecare cu un număr par de noduri, în care fiecare nod are maxim 2 fii. Numărul maxim de noduri de pe ultimul nivel **i**, știind că rădăcina arborelui se află pe nivelul **1**, este?

- a) 2^{i+1} b) 2^{i+1} c) 2^i d) $2^{\frac{i+1}{2}}$ e) 2^{i-1} f) $2^{i-1} - 1$

13. Un graf este bipartit dacă nodurile lui pot fi împărțite în două submulțimi disjuncte, astfel încât fiecare muchie are o extremitate în prima submulțime și cealaltă în a doua submulțime. Fie **G** un graf neorientat, bipartit, cu **10** noduri. Numărul maxim de muchii pe care poate să le aibă graful **G** este?

- a) 5 b) 15 c) 25 d) 35 e) 45 f) 55

14. Complexitatea următoarei funcții este?

Limbajul C/C++ <pre> void f(int n, int v[101]) { int i, j=0; for(i=0; i<n; i++) while(j<n && v[i]<v[j]) j++; } </pre>	Limbajul Pascal <pre> procedure f(n:integer; v:array[0..100] of integer); var i,j:integer; begin j:=0; for i:=0 to n-1 do while (j<n) and (v[i]<v[j]) do inc(j) end; end; </pre>
--	--

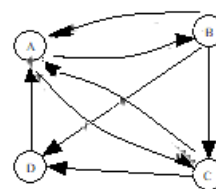
- | | | |
|------------------|-----------|------------------|
| a) $O(\sqrt{n})$ | b) $O(n)$ | c) $O(n \log n)$ |
|------------------|-----------|------------------|

e) $O(n(\log n)^2)$

d) $O(n^2)$

f) $O(n^3)$

15



Numărul de drumuri de lungime 3 din graful orientat alăturat este?

a) 10

b) 15

c) 20

d) 25

e) 30

f) 40

Varianta 46

1. Variabila întreagă **v** memorează numere naturale. Valoarea variabilei **v** după rularea următoarei secvențe de instrucțiuni este?

Limbaajul C/C++

```
v=2; v=v*v; v=v-v%2*3;
v=v%3+5;
```

Limbaajul Pascal

```
v:=2; v:=v*v;
v:=v-v mod 2*3;
v:=v mod 3 + 5;
```

- a) 4 b) 5 c) 6 d) 7 e) 8 f) 9

2. Variabilele întregi **i**, **x** memorează numere naturale. Numărul de numere de 5 cifre ce vor fi afișate după rularea următoarei secvențe de instrucțiuni este?

Limbaajul C/C++

```
for(i=1;i<=100000;i++){
    x=i-5;
    while(x>2&&(x%10==0||x%10==2)){
        x=x/10,x=x-5;
    }
    if(x==0||x==2)
        cout<<i;
    printf("%d",i);
}
```

Limbaajul Pascal

```
for i:=1 to 100000 do
begin
    x:=i-5;
    while (x>2)and((x mod
10=0) or (x mod 10=2)) do
begin
        x:=x div 10; x:=x-5
    end;
    if (x=0) or (x=2) then
        write(i)
    end;
```

- a) 28 b) 29 c) 30 d) 31 e) 32 f) 33

3. După rularea următoarei secvențe de instrucțiuni se afișează?

Limbaajul C/C++

```
int i=0,j=5,aux,
    v[]={41,52,26,11,48,65};
while(i<j){
    for(;i<j && !(v[i]%2);i++);
    for(;i<j && (v[j]%2);j--);
    if(i<j)
        aux=v[i],v[i]=v[j],v[j]=aux;
}
for(i=0;i<6;++i)
    cout<<v[i]<<" ";
    printf("%d ",v[i]);
```

Limbaajul Pascal

```
var i,j,aux:integer;
    v:array[0..5] of
integer= (41, 52, 26,11,
48, 65);
i:=0; j:=5;
while i<j do begin
    while (i<j)and(not(v[i]
mod 2=1)) do
        inc(i);
    while (i<j)and(v[j] mod
2=1) do dec(j);
    if i<j then begin
        aux:=v[i];
        v[i]:=v[j];
        v[j]:=aux end;
    end;
```

```
for i:=0 to 5 do
  write(v[i], ' ');
```

- a) 11 26 41 48 52 65 b) 65 41 11 52 48 26
 c) 26 48 52 11 41 65 d) 48 52 26 11 41 65
 e) 52 48 26 65 41 11 f) 11 41 65 48 52 26

4. După rularea următoarei secvențe de instrucțiuni se afișează?

Limbajul C/C++

```
char s[50]="test informatica";
char *p;
strtok(s, " ");
p=strtok(NULL, " ");
strcpy(s, strcat(p, s));
cout<<s; | printf("%s", s);
```

Limbajul Pascal

```
var s,p:string;
s:='test informatica';
p:=copy(s,pos(' ',s)+1,
length(s) - pos(' ',s));
delete(s,pos(' ',s),
length(s)-pos(' ',s)+1);
s:=p+s; write(s);
```

- a) testtest b) testinformatica c) test
 d) informaticatest e) informatica f) info

5. Pentru ca secvența următoare de instrucțiuni să determine dacă cele n intervale memorate în tabloul unidimensional v sunt disjuncte, punctele de suspensie se înlocuiesc cu?

Limbajul C/C++

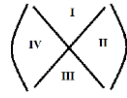
```
struct interval{int x, y;}
v[100];
int i, j, r, t, n;
r=v[0].x; t=v[0].y;
for(i=1; i<n; ++i) {
  if(r<v[i].x) r=v[i].x;
  if(t>v[i].y) t=v[i].y;
  if(...)
  cout<<"DA"; | printf("DA");
  else
  cout<<"NU"; | printf("NU");
```

Limbajul Pascal

```
type interval=record
  x,y:integer end;
var v:array[0..99] of
  interval;
  i,j,r,t,n:integer;
r:=v[0].x; t:=v[0].y;
for i:=1 to n-1 do begin
  if r<v[i].x then
    r:=v[i].x;
  if t>v[i].y then
    t:=v[i].y
end;
if ... then write('DA')
else write('NU');
```

- a) $r < t$ b) $r \leq t$ c) $r == t$ d) $r != t$ | $r < > t$ e) $r \geq t$ f) $r > t$

6. Variabilele întregi i , j , s memorează numere naturale. Tabloul bidimensional a , pătratic, cu elemente numere naturale, are n linii și n coloane, numerotate de la 1 la n și



este împărțit în 4 zone ca în desen. Pentru ca algoritmul următor să adune elemente din zona II, punctele de suspensie se înlocuiesc cu?

Limbajul C++/C

```
s=0;
for(i=1;i<=(n-1)/2;++i)
    for(j=i+1;j<=n-i;j++)
        s+=.....;
```

a) $a[n-i+1][j]$ | $a[n-i+1, j]$

c) $a[n+1-i][n+1-j]$ |
 $a[n+1-i, n+1-j]$

e) $a[j][i]$ | $a[j, i]$

Limbajul Pascal

```
s:=0;
for i:=1 to (n-1) div 2 do
    for j:=i+1 to n-i do
        s:=s+.....;
```

b) $a[i][j]$ | $a[i, j]$

d) $a[n-i][n-j]$ | $a[n-i, n-j]$

f) $a[n+1-j][n+1-i]$ |
 $a[n+1-j, n+1-i]$

7. Pentru ca, după rularea următoarei secvențe de instrucțiuni, să se afișeze valoarea 7, punctele de suspensie se înlocuiesc cu?

Limbajul C/C++

```
int i, p, v[10]={2, 6, 8, 12,
20, 25, 30, 37, 41, 92};
for (p = 1; p < 10; p *=2);
for (i = 0; p; ...)
    if (i+p<10 && v[i+p]<= 40)
        i += p;
cout<<i; | printf("%d", i);
```

a) $p/=2$ | $p \text{ div } 2$

d) $p--$ | $p-1$

b) $p*2$

e) $p+2$

Limbajul Pascal

```
var v:array[0..9] of
integer = (2, 6, 8, 12,
20, 25, 30, 37, 41, 92);
i,p:integer;
p:=1;
while p<10 do p:=p*2;
i:=0;
while p<>0 do begin
    if (i+p<10) and
(v[i+p]<=40) then
        i:=i+p;
        p:=.....;
end;
write(i);
```

c) $p++$ | $p+1$

f) $p+=i$ | $p+i$

8. După rularea următorului program se afișează?

Limbajul C++

```
#include<iostream>
using namespace
std;
```

Limbajul C

```
#include <stdio.h>
struct Q{
    int a, b, c;};
```

Limbajul Pascal

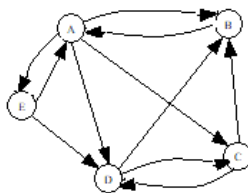
```
type QT = record
a,b,c:integer
end;
```

<pre> struct Q{ int a, b, c;}; struct S{ int a, b, c; struct Q Q;}; int main(){ Q Q={3, 2, 1}; S S={4, 5, 6}; S.Q=Q; cout<<S.b-S.Q.b; return 0;} </pre>	<pre> struct S{ int a, b, c; struct Q Q;}; int main() { struct Q Q={3,2,1}; struct S S={4,5,6}; S.Q=Q; printf("%d", S.b-S.Q.b); return 0;} </pre>	<pre> ST = record a,b,c:integer; Q:QT end; var Q:QT = (a:3;b:2;c:1); S:ST = (a:4;b:5;c:6); begin S.Q:=Q; write(S.b-S.Q.b) end. </pre>			
a) 5	b) 4	c) 3	d) 2	e) 1	f) 0

9. Apelul $f(6,2)$ are valoarea?

Limbajul C/C++ <pre> int f(int x, int y) { if(x>y) return f(f(y,x), x/y)-1; else if(x==y) return f(x+y, x)+2; else return y-x;} </pre>	Limbajul Pascal <pre> function f(x,y:integer): integer; begin if x>y then f:=f(f(y,x),x div y)-1 else if x=y then f:=f(x+y,x)+2 else f:=y-x; end; </pre>				
a) 0	b) 1	c) 2	d) 3	e) 4	f) 5

10. Numărul de circuite elementare diferite (care au cel puțin un arc diferit) care trec prin



nodul **A** din graful orientat alăturat este?

a) 4 b) 5 c) 6 d) 7 e) 8 f) 9

11. Folosind metoda backtracking, se determină în ordine lexicografică, toate cuvintele de 4 litere distincte din mulțimea {a,b,e,f,g,i,l,m,o,p,r,u} oricare două litere alăturate să nu fie consecutive în mulțimea dată, iar vocalele să apară în ordine descrescătoare. Primele 5 soluții sunt: **afbg**, **afbl**, **afbm**, **afbp**, **afbr**. Cele 2 cuvinte generate înaintea cuvântului **elaf** sunt?

a) farg farl	b) egrl egrm	c) egmf egmp
d) eglr eglu	e) buri buro	f) bopl bupm

12. Un arbore oarecare cu rădăcină este reprezentat prin vectorul de tați t . Dacă algoritmul următor determină nivelul pe care se găsește un nod x în arbore, cu ce secvență de cod se pot înlocui punctele de suspensie de mai jos?

Limbajul C/C++

```
int nivel=0;
while(t[x]){
    ...
    nivel++;}
```

Limbajul Pascal

```
nivel:=0;
while t[x]<>0 begin
    ...
    inc(nivel)
end;
```

- a) $t[x]=x$; | $t[x]:=x$; b) $t[x]--$; | $t[x]:=t[x]-1$;
c) $x=t[x]$; | $x:=t[x]$; d) $t[x]++$; | $t[x]:=t[x]+1$;
e) $x=t[t[x]]$; | $x:=t[t[x]]$; f) $t[x]=x+t[x]$; | $t[x]:=x+t[x]$;

13. Numărul de numere întregi din intervalul $[100, 10000]$ pentru care rularea următoarei secvențe de instrucțiuni afișează valoarea 5 este?

Limbajul C/C++

```
cin>>n; | scanf("%d",&n);
while(n>9) n=n/10+n%10;
cout<<n; | printf("%d",n);
```

Limbajul Pascal

```
readln(n);
while n>9 do n:=n div 10
+ n mod 10;
write(n);
```

- a) 1100 b) 1110 c) 1200 d) 1450 e) 1500 f) 1890

14. Un algoritm determină minimul și maximul dintr-un tablou unidimensional cu 100 de numere, prin operații de comparare a elementelor. Numărul minim de comparații necesare este?

- a) 140 b) 142 c) 144 d) 146 e) 148 f) 150

15. În graful neorientat G cu 100 de noduri, două noduri i și j sunt adiacente dacă $|i - j| = 8$ sau $|i - j| = 12$. Numărul de componente conexe ale grafului este:

- a) 1 b) 2 c) 3 d) 4 e) 25 f) 50

INDICAȚII ȘI RĂSPUNSURI

Varianta 1

Indicații și răspunsuri

1. **Răspuns corect:**

a) not ((a<-3) or (a>2)) or (a=3) or (a=5) or (a=9) (Pascal) respectiv
 !((a<-3) || (a>2)) || (a==3) || (a==5) || (a==9) (C/C++)

Indicații: Valoarea lui **a** trebuie să fie mai mare sau egală cu -3 și mai mică sau egală decât 2 sau egală cu 3 sau egală cu 5 sau egală cu 9

2. **Răspuns corect:** c) $q-1+j$

3. **Răspuns corect:** f) info

4. **Răspuns corect:** d)

Limbajul C++/Limbajul C

```
s=0; i=1;
while(i<=n)
{ s=s+x[p][i];
  i++; }
```

Limbajul Pascal

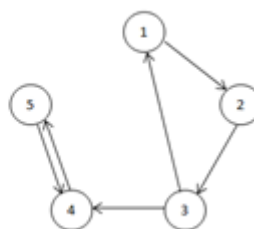
```
s:=0; i:=1;
while i<=n do
begin
  s:=s+x[p,i];
  i:=i+1;
end;
```

Indicații:

Fiind vorba de suma elementelor pe linia **p**, primul indice al elementului din tabloul bidimensional trebuie să fie **p**. Având **n** coloane în matrice, cel de-al doilea indice al elementului din matrice trebuie să parcurgă toate valorile de la **1** la **n**.

5. **Răspuns corect:**

a) $\begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{matrix}$



Indicații: Se observă cu ajutorul desenului de mai sus, că de la nodurile 4 și 5 nu există niciun drum la nodurile 1, 2 și 3. Deci în matrice doar elementele a_{41} , a_{42} , a_{43} , a_{51} , a_{52} și a_{53} vor avea valoarea 0 restul elementelor din matrice având valoarea 1.

6. **Răspuns corect:** b) (3, 3, 3)

Indicații: Mulțimea **A₂** nu conține elementul 3, deci în produsul cartezian a celor 3 mulțimi nu putem avea ca soluție (3, 3, 3).

7. **Răspuns corect:** a)

Limbajul C++/LimbajulC

```
a) if ( p(x,x)==2) cout<<"prim";  
    | printf("prim");
```

Limbajul Pascal

```
a) if p(x, x)=2 then  
    write('prim');
```

Indicații: Funcția calculează numărul de divizori al lui a care sunt mai mici sau egali cu b.

8. **Răspuns corect:** d) 2, 4, 0, 3, 4

9. **Răspuns corect:** f) 673656

Indicații: În urma apelului $t(7,7)$ $x=7-1=6$, $y=6+1=7$ deci se afișează 67, iar la ieșirea din funcție y va avea valoarea 6 datorită primului parametru al funcției transmis prin adresă, iar x va avea valoarea 3. Se va afișa x și y adică 3 și 6. În urma apelului $t(6,3)$ $x=6-1=5$, $y=5+1=6$, se va afișa 56.

10. **Răspuns corect:** c)

Limbajul C++/LimbajulC

```
c) par(a, b) == (b-a+1)/2
```

Limbajul Pascal

```
c) par(a, b) = (b-a+1) DIV 2
```

Indicații: Pentru verificarea expresiilor, se pot folosi perechi de valori cu aceeași paritate sau de parități diferite.

11. **Răspuns corect:** c)

Limbajul C++/LimbajulC

```
c) c = mini(c, v[i]);
```

Limbajul Pascal

```
c) c :=mini(c, v[i]);
```

12. **Răspuns corect:** e) amTre

13. **Răspuns corect:** a)

Limbajul C++/LimbajulC

```
a) e[10] = x;
```

Limbajul Pascal

```
a) e[10] := x;
```

14. **Răspuns corect:** f) 5120

Indicații: Cu 5 noduri se pot forma $2^{(5*4)/2}=2^{10}=1024$ grafuri neorientate distincte. Deci avem 1024 de grafuri distincte în care nodul 2 este adiacent cu nodul 1, alte 1024 de grafuri distincte în care nodul 2 este adiacent cu nodul 3, ..., 1024 de grafuri distincte în care nodul 2 este adiacent cu nodul 6. În total sunt $1024*5=5120$ de grafuri distincte.

15. **Răspuns corect:** c) 5

Varianța 2

Indicații și răspunsuri

1. **Răspuns corect:** c) 543212222543445

2. **Răspuns corect:** d)

Limbajul C++/Limbajul C

d) $(x > y \mid \mid x < z) \ \&\& \ x > t$

Limbajul Pascal

d) $((x > y) \text{ or } (x < z)) \text{ and } (x > t)$

3. **Răspuns corect:** e)

Limbajul C++/Limbajul C

e) $a[i][j] == a[n+1-j][n+1-i]$

Limbajul Pascal

e) $a[i, j] = a[n+1-j, n+1-i]$

Indicații: Se observă că elementele unui tablou bidimensional cu 4 linii și 4 coloane sunt simetrice față de diagonala secundară astfel: $a_{11}=a_{44}$, $a_{12}=a_{34}$, $a_{13}=a_{24}$, $a_{21}=a_{43}$, $a_{22}=a_{33}$, $a_{31}=a_{42}$, deci putem deduce expresia pe caz general $a_{i,j}=a_{n+1-j,n+1-i}$.

4. **Răspuns corect:** d) 10

5. **Răspuns corect:** f) $(s.A.y + s.B.y) / 2$

6. **Răspuns corect:** b) 0010

0100

1000

0001

Indicații: Se observă că fiecărui tabloul bidimensional îi corespunde o permutare. Permutarea este o funcție $f: \{1,2,\dots,n\} \rightarrow \{1,2,\dots,n\}$, bijectivă. Notăm linia 1000 cu 1, linia 0100 cu 2, linia 0010 cu 3 și linia 0001 cu 4. Observăm ca permutarea pentru tabloul bidimensional dat este (3,1,4,2) iar permutarea următoare acesteia din punct de vedere lexicografic este (3,2,1,4), corespunzătoare tabloului bidimensional de la punctul b.

7. **Răspuns corect:** d) 3

Indicații: Funcția calculează numărul de moduri distincte în care poate fi scris un număr x ca sumă de y numere naturale strict pozitive. Deci 6 poate fi scris în 3 moduri: $6=1+2+3$, $6=1+1+4$, $6=2+2+2$, același rezultat se obține urmărind apelurile recursive efectuate de funcția nr.

8. **Răspuns corect:** b) 5, 6, 7, 9, 10

9. **Răspuns corect:** e) 2 2

Indicații: În urma apelului $F(x, y)$ respectiv $F(\&x, y)$, variabila x este singura variabilă care își modifică valoarea după apel, fiind un parametru transmis prin adresă, deci y rămâne neschimbat.

10. **Răspuns corect:** c) 3

Indicații: O soluție: culoarea 1 nodurile 1, 4 și 6, culoarea 2 nodurile 3 și 8, culoarea 3 nodurile 5 și 7, iar nodul 2 fiind nod izolat propun culoarea 1 (se putea colora și cu 2 și cu 3).

11. **Răspuns corect:** f) 45

Indicații: Avem formula: $N_{\text{maxmuchii}} = (n-p)(n-p+1)/2$, unde **n** reprezintă numărul de noduri iar **p** numărul de componente conexe.

12. **Răspuns corect:** a)

Limbaajul C++/Limbaajul C

```
a) cout<<strchr(c, 'd')-c; |  
    printf("%d", strchr(c, 'd')-c );
```

Limbaajul Pascal

```
a) write(pos('d', c));
```

13. **Răspuns corect:** c) 2021

Indicații: Instrucțiunea for nu produce efecte

14. **Răspuns corect:** e) combinărilor de 30 de elemente luate câte 5

Indicații: Nu contează ordinea în echipă

15. **Răspuns corect:** f) 14

Varianta 3

Indicații și răspunsuri

1. **Răspuns corect:** e) **64**

Indicații: n poate avea valori de la **36** la **99**

2. **Răspuns corect:** a)

3. **Răspuns corect:** d)

Indicații: Suma tuturor elementelor tabloului a este **0**. În cadrul programului, se adună toate elementele tabloului a , mai puțin cele pentru care $i+1=j$. Sunt **19** elemente pentru care $i+j=1$ și fiecare dintre ele are valoarea **-1**. Deci $s+(-19)=0 \Rightarrow s=19$.

4. **Răspuns corect:** d)

Indicații: Algoritmul nu este corect implementat. În majoritatea cazurilor, generează ciclare infinită.

5. **Răspuns corect:** d)

Indicații: Se ține cont de ordinea și modul de transmitere al parametrilor

6. **Răspuns corect:** b)

Indicații: Funcția dată calculează $x*y$

7. **Răspuns corect:** a)

Indicații: Se verifică dacă elementul are loc în stivă

8. **Răspuns corect:** f)

9. **Răspuns corect:** e)

Indicații: Numărul poate avea **1, 2** sau **3** cifre

10. **Răspuns corect:** c)

11. **Răspuns corect:** e)

Indicații: Numărul valorilor se calculează direct prin formula $n*(n-1)/2$

12. **Răspuns corect:** c)

13. **Răspuns corect:** a)

Indicații: Arcele $(2,1)$, $(2,3)$ și $(2,4)$ au extremitatea inițială nodul **2** (cu gradul exterior **0**), iar arcele $(3,1)$ și $(4,1)$ au extremitatea finală nodul **1** (cu gradul interior **0**)

14. **Răspuns corect:** b)

Indicații: Numărul ciclurilor hamiltoniene dintr-un graf complet cu n noduri este:
 $(n-1)!/2$.

15. **Răspuns corect:** d)

Indicații: Afirmările **1**, **4** și **5** sunt adevărate.

Varianta 4

Indicații și răspunsuri

1. **Răspuns corect:** d)
Indicații: Îndeplinesc condiția cerută expresiile 1, 2 și 4
2. **Răspuns corect:** f)
Indicații: două drepte paralele au aceeași pantă
3. **Răspuns corect:** e) $a+b+c-f(a, f(c, b))$
Indicații: $a+b+c-\max(a, b, c) = a+b+c-\max(a, \max(b, c))$
4. **Răspuns corect:** a) -1 2 2 -1 -1 1
Indicații: Mecanismul de transmiterea parametrilor.
5. **Răspuns corect:** e) 152004
Indicații: Se mută grupurile de câte 2 cifre. Datorită numărului impar de cifre a lui n , în m apare o cifră în plus.
6. **Răspuns corect:** d) 10239
Indicații: Algoritmul determină baza minimă x în care îl consideră pe n -ul inițial și îl transformă în baza 10
7. **Răspuns corect:** e)
Indicații: Linia $n-j+1$ din matricea a devine coloana j în matricea b
8. **Răspuns corect:** c)
9. **Răspuns corect:** b)
Indicații: Indicii sunt de la 1 la n ; parcurgere liniară a vectorului.
10. **Răspuns corect:** f) Inserează șirul t în șirul s , începând cu poziția k
11. **Răspuns corect:** a) 0
Indicații: Graful este tare conex
12. **Răspuns corect:** f) (1, 3, 5, 2, 1, 2)
Indicații: (3, 2, 1, 5, 1, 1) – are număr impar de noduri de grad impar
(5, 1, 6, 4, 5, 3) și (1, 1, 1, 1, 1, 6) – au un nod cu grad prea mare
(1, 1, 1, 1, 2, 2) – nu poate fi conex deoarece are doar 4 muchii
(2, 1, 3, 1, 0, 1) are un nod izolat
13. **Răspuns corect:** b) 8

Indicații: $3n+2 \leq n(n-1)/2$

14. **Răspuns corect:** c) 3

Indicații: Numerele 12, 16 și 18 generează arbori cu 8 frunze

15. **Răspuns corect:** d) 25

Indicații: $C_5^2 + C_5^3 + C_5^4$

Varianța 5

Indicații și răspunsuri

1. **Răspuns corect:** e)
2. **Răspuns corect:** b)
Indicații: Operatorul % (mod) nu funcționează pe tipul real
3. **Răspuns corect:** c) `dmtr`
Indicații: La ștergerea unei litere, vecina din dreapta îi va lua locul și nu va mai fi eliminată
4. **Răspuns corect:** d) Graful **G** conține cel puțin un ciclu
Indicații: Graful aciclic maximal cu **100** de noduri este un arbore și are **99** de muchii.
5. **Răspuns corect:** d) **30**
Indicații: Numarul de valori **1** din matricea de adiacență este egal cu numărul de arce
6. **Răspuns corect:** e) **5**
Indicații: Nodurile terminale au gradul **1**, restul nodurilor având gradul **3**
7. **Răspuns corect:** b)
8. **Răspuns corect:** a) (1, 21, 13, 23, 33, 17, 27)
Indicații: Șirul trebuie să fie sortat în funcție de cifra unităților
9. **Răspuns corect:** c)
10. **Răspuns corect:** b) **12**
11. **Răspuns corect:** a) **3** și **3**
Indicații: Parametrul transmis prin valoare nu se modifică, pe când cel transmis prin referință/adresă, da.
12. **Răspuns corect:** a)
Indicații: Atribuirea este corectă între două variabile de același tip RECORD/struct
13. **Răspuns corect:** e) **15**
Indicații: Matricea de adiacență are valori **0** pe diagonala principală. Dintre celelalte **6** elemente, **4** trebuie să conțină valori **1**, ordinea nefiind importantă. Așadar numărul de grafuri este C_6^4 .
14. **Răspuns corect:** e)
15. **Răspuns corect:** b) Graful **G** este un graf hamiltonian

Varianta 6

Indicații și răspunsuri

1. **Răspuns corect:** b) 8
Indicații: numărul este par ($k*2$).
2. **Răspuns corect:** c) $(a+b)\%2==0$
Indicații: suma a două numere de aceeași paritate este un număr par.
3. **Răspuns corect:** a) aticamatica
4. **Răspuns corect:** b) 9
Indicații: se obține un nod izolat
5. **Răspuns corect:** d) (1,2,2,1,2,2)
Indicații: suma gradelor tuturor nodurilor unui graf neorientat este un număr par.
6. **Răspuns corect:** b)12
7. **Răspuns corect:** b) 3
Indicații: graful va conține un circuit elementar cu toate nodurile grafului și încă două arce care au extremitatea inițială în același nod.
8. **Răspuns corect:** b) Suma elementelor de pe diagonala secundară a tabloului a
Indicații: pentru ca un element să se găsească pe diagonal secundară trebuie ca suma dintre indicele liniei și cel al coloanei să fie $n+1$.
9. **Răspuns corect:** b) 4
Indicații: $x=a[1][2]+a[3][4]=1+3=4$
10. **Răspuns corect:** a) aranjamentelor
Indicații: elementele ce formează o soluție sunt distincte iar ordinea lor în cadrul unei soluții contează.
11. **Răspuns corect:** d) 60
Indicații: $C_5^2 * C_4^2$
12. **Răspuns corect:** c) determinarea elementului maxim din șir

13. **Răspuns corect:** a) $p.x * p.y > 0$
Indicații: în aceste cadrane abscisa și ordonata au același semn.
14. **Răspuns corect:** b) 90
Indicații: toate nodurile îl au ca “tată” pe nodul etichetat cu 10.
15. **Răspuns corect:** b) 2

Varianta 7

Indicații și răspunsuri

1. **Răspuns corect:** b) $!(x < -4 \parallel x > -1) \parallel !(x < 1 \parallel x > 4) \parallel !(x < 10)$ (C++/C)
respectiv b) $\text{not}((x < -4) \text{ or } (x > -1)) \text{ or } \text{not}((x < 1) \text{ or } (x > 4)) \text{ or } \text{not}(x < 10)$ (Pascal)
2. **Răspuns corect:** c) $\text{floor}(5.19) == \text{floor}(5.91)$ (C++/C)
respectiv c) $\text{trunc}(5.19) = \text{trunc}(5.91)$ (Pascal)
3. **Răspuns corect:** d) (3, 4, 10, 17, 46)
4. **Răspuns corect:** e) 3 (valorile comparate cu x fiind 12, 18, 17)
5. **Răspuns corect:** f) 3 (5, 6, 7)
6. **Răspuns corect:** d) 20 ($C_6^3 = 20$)
7. **Răspuns corect:** e) 6
8. **Răspuns corect:** f) `f.close();` (C++) sau `fclose(f);` (C) sau `close(f);` (Pascal)
9. **Răspuns corect:** c) 9
10. **Răspuns corect:** f) $((c.p1 + c.p2) * 0.8 + c.medbac * 0.2) \geq 5.0$
11. **Răspuns corect:** d) `void cifre (unsigned n, unsigned &prim, unsigned &ult)` (C++)
`void cifre (unsigned n, unsigned *prim, unsigned *ult)` (C)
`procedure cifre (n: longint; var prim, ult: byte);` (Pascal)
12. **Răspuns corect:** f) 10 și 1
13. **Răspuns corect:** c) 15 și 210 (nr minim arce = nr vârfuri; nr maxim = $n * (n - 1)$)
14. **Răspuns corect:** f) 1 și 2
15. **Răspuns corect:** d) 63

Varianta 8

Indicații și răspunsuri

1. **Răspuns corect:** e) $!((x >= -4 \ \&\& \ x <= -1) \ || \ (x >= 1 \ \&\& \ x <= 4) \ || \ (x >= 10))$ (C/C++)
respectiv e) $\text{not}((x >= -4 \ \text{and} \ x <= -1) \ \text{or} \ (x >= 1 \ \text{and} \ x <= 4) \ \text{or} \ (x >= 10))$ (Pascal)
2. **Răspuns corect:** c) $\text{floor}(5.19) == \text{floor}(5.91)$ (C/C++)
respectiv c) $\text{trunc}(5.19) = \text{trunc}(5.91)$ (Pascal)
3. **Răspuns corect:** f) (3, 4, 7, 10, 12, 17, 18, 20, 46)
4. **Răspuns corect:** e) (3, 4, 10, 17, 46) și (7, 10, 12, 18, 20)
5. **Răspuns corect:** c) 9
6. **Răspuns corect:** d) 70
7. **Răspuns corect:** e) 512
8. **Răspuns corect:** d) $f.\text{close}();$ (C++) sau $\text{fclose}(f);$ (C) sau $\text{close}(f);$ (Pascal)
9. **Răspuns corect:** b) '9'
10. **Răspuns corect:** f) $(e.\text{sex} == 'F' \ || \ e.\text{sex} == 'f') \ \&\& \ (e.\text{dn.l} == 7 \ \&\& \ e.\text{dn.z} <= 10)$ (C/C++)
respectiv $((e.\text{sex} == 'F') \ \text{or} \ (e.\text{sex} == 'f')) \ \text{and} \ (e.\text{dn.l} == 7) \ \text{and} \ (e.\text{dn.z} <= 10)$ (Pascal)
11. **Răspuns corect:** c) $\text{return suma}(n);$ (C/C++)
respectiv c) $\text{suma}(n);$ (Pascal)
12. **Răspuns corect:** f) 56 (numărul minim de muchii se obține pentru două componente conexe, fiecare graf complet cu 8 vârfuri; $m = 8 * 7 / 2 = 28$ reprezintă numărul de muchii pentru fiecare dintre aceste componente)
13. **Răspuns corect:** e) 9 (graful, având 10 vârfuri, gradul maxim al unui vârf poate fi $n - 1 = 9$)
14. **Răspuns corect:** c) Graful este (slab) conex
15. **Răspuns corect:** d) 33

Varianta 9

Indicații și răspunsuri

1. **Răspuns corect: b)**

Indicații: verificare directă a fiecărei variante folosind pentru n o valoare cu cel puțin 4 cifre

2. **Răspuns corect: f)**

Indicații: se ține cont de precedența operatorilor și de regulile de negare a unei expresii logice

3. **Răspuns corect: d)**

Indicații: șirul inițial are 8 caractere, pentru că la fiecare iterație se elimină un caracter. Mai departe, se analizează caracterele afișate : primul caracter este 'a', prin urmare al doilea caracter din șirul inițial trebuie să fie 'b' ș.a.m.d

4. **Răspuns corect: d)**

Indicații: un arbore cu 4 noduri are 3 muchii, este conex și fără cicluri

5. **Răspuns corect: d)**

Indicații: calcul direct al valorilor din matrice, apoi se numără câte dintre elemente sunt valori prime

6. **Răspuns corect: b)**

Indicații: se analizează soluțiile din enunț: soluția care are pe prima poziție valoarea 1 va fi prima generată, apoi urmează soluțiile care au pe prima poziție valoarea 2; soluția care conține valoarea 3 pe poziția a 2-a va fi generată înaintea celei care conține pe poziția a 2-a valoarea 8; soluția care are pe prima poziție valoarea 3 va fi generată ultima

7. **Răspuns corect: e)**

Indicații: (1,2,3,1), (1,3,5,1), (3,4,5,3), (1,2,3,5,1), (1,3,4,5,1), (1,2,3,4,5,1)

8. **Răspuns corect: d)**

Indicații: componentele tare conexe sunt alcătuite din următoarele mulțimi de noduri {1,2,3,6},{4},{5},{7}

9. **Răspuns corect: e)**

Indicații: lanțurile de lungime maximă se află între nodurile 9 și 12, 9 și 8, 9 și 11, 10 și 12, 10 și 8, 10 și 11

10. **Răspuns corect: e)**

Indicații: se verifică dacă distanța de la centrul cercului la originea sistemului de coordonate este mai **mică** decât raza cercului

11. **Răspuns corect: a)**

12. **Răspuns corect: a)**

13. **Răspuns corect: b)**

Indicații: variabila globală y este vizibilă în toate funcțiile și va fi modificată de fiecare apel al acestora. Variabila globală x nu va fi modificată, deoarece în fiecare dintre funcții există un parametru cu același nume, iar apelul $g(x)$ va prelua pe segmentul de stivă doar valoarea lui x .

14. **Răspuns corect: d)**

Indicații: secvența determină numărul de numere cu cel mult 4 cifre care au numărul divizorilor număr par; se știe că doar pătratele perfecte au număr impar de divizori

15. **Răspuns corect: e)**

Indicații: funcția verifică (utilizând Divide et Impera) dacă vectorul este sortat strict crescător între pozițiile 2 și 5

Varianta 10

Indicații și răspunsuri

1. **Răspuns corect: c)**

Indicații: verificare directă, ținând cont de precedența operatorior

2. **Răspuns corect: d)**

Indicații: verificare directă a fiecărei variante de răspuns

3. **Răspuns corect: b)**

Indicații: Algoritmul determină ultima cifră a numărului x^y . Cum ultima cifră a lui x este 7, ultima cifră a puterilor lui x va fi 7, 9, 3, 1, apoi se repetă. Pentru y sunt posibile 90 de valori (de la 10 la 99), înseamnă că 22 dintre acestea vor determina pentru z valoarea 1.

4. **Răspuns corect: d)**

5. **Răspuns corect: a)**

Indicații: $f(f(775125) + f(97917)) = f(5+7) = f(12) = -1$

6. **Răspuns corect: f)**

Indicații: primele 11 soluții sunt: $\{1\}$, $\{1,2\}$, $\{1,2,3\}$, $\{1,2,3,4\}$, $\{1,2,3,4,5\}$, $\{1,2,3,4,5,6\}$, $\{1,2,3,4,5,6,7\}$, $\{1,2,3,4,5,7\}$, $\{1,2,3,4,6\}$, **$\{1,2,3,4,6,7\}$** , **$\{1,2,3,4,7\}$**

7. **Răspuns corect: e)**

Indicații: nodurile 1 și 3 au gradul 3 și sunt adiacente, prin urmare pentru a obține un graf eulerian este necesar ca acestea să ajungă să aibă gradul 4.

8. **Răspuns corect: b)**

Indicații: cele 16 muchii pot determina o componenta conexă cu 17 noduri (arbore), deci rămân 13 noduri izolate și prin urmare numărul minim de componente conexe este 14. Pentru a determina numărul maxim de componente conexe, se caută graful complet cel mai mare care are cel mult 16 muchii. K_6 are 15 muchii, deci vom avea o componentă conexă cu 7 noduri, 23 de noduri izolate, adică 24 componente conexe

9. **Răspuns corect: c)**

Indicații: graful conține arcele: (2,4), (2,6), (2,8), (2,10), (3,6), (3,9), (4,8), (5,10)

10. **Răspuns corect: b)**

11. **Răspuns corect: c)**

Indicații: fiecare apel recursiv va gestiona propria variabilă locală i

12. **Răspuns corect: d)**

Sunt 12 valori afișate: 19 20 21 22 30 38 46 45 44 43 35 27

13. **Răspuns corect: f)**

Indicații: verificare directă

14. **Răspuns corect: d)**

Indicații: după executarea subprogramului, vectorul a conține valorile (1,6,15,20,15,6,1)

15. **Răspuns corect: e)**

Indicații: secvența determină suma cifrelor în baza $b=3$. Pentru a determina suma maximă, încercăm să considerăm cât mai multe cifre 2 (cifra maximă în baza 3):
 $2 \cdot 1 + 2 \cdot 3 + 2 \cdot 9 + 2 \cdot 27 = 80$

Varianta 11

Indicații și răspunsuri

1. **Răspuns corect e)**

2. **Răspuns corect c)**

3. **Răspuns corect f)**

Indicație: Este suficient să fie calculat radical de ordinul 3 din n pentru a afla câte cuburi mai mici sau egale decât n există.

4. **Răspuns corect d)**

5. **Răspuns corect e)**

6. **Răspuns corect c)**

7. **Răspuns corect a)**

Indicație: Pentru ca un graf neorientat cu n noduri să fie conex, numărul minim de muchii necesare este $n-1$.

8. **Răspuns corect f)**

9. **Răspuns corect d)**

Indicație: Suma gradelor nu trebuie să fie egală cu $2n-2$ (n =numărul vârfurilor)

10. **Răspuns corect e)**

11. **Răspuns corect b)**

Indicație: La sumă se adună doar numerele divizibile cu 3, iar condiția de oprire a recursivității este când i este egal cu $3*n$.

12. **Răspuns corect b)**

13. **Răspuns corect a)**

Indicație: pentru $b=29$ și $b=30$.

Subprogramul calculează numărul valorilor naturale impare din intervalul $[a, b]$.

14. **Răspuns corect b)**

Indicație: Calculează C_n^k

15. **Răspuns corect b)**

Indicație: Calculează ultima cifră a lui x^y

Varianta 12

Indicații și răspunsuri

1. **Răspuns corect b)**

Indicație: numărul muchiilor unui graf neorientat complet este $n*(n-1)/2$

2. **Răspuns corect a)**

Indicație: condiția ca un element să se situeze pe diagonala secundară a unui tablou bidimensional este $i+j=n+1$ (unde i, j reprezintă indicii de linie și coloană ai elementului)

3. **Răspuns corect d)**

4. **Răspuns corect f)**

5. **Răspuns corect a)**

6. **Răspuns corect e)**

7. **Răspuns corect c)**

8. **Răspuns corect d)**

9. **Răspuns corect a)**

Indicație: Suma gradelor trebuie sa fie egala cu $2n-2$ (n =numărul vârfurilor)

10. **Răspuns corect d)**

11. **Răspuns corect b)**

12. **Răspuns corect e)**

13. **Răspuns corect c)**

14. **Răspuns corect d)**

Indicație: Valoarea calculată reprezintă numărul divizorilor pozitivi ai variabilei n

15. **Răspuns corect c)**

Varianta 13

Indicații și răspunsuri

1. **Răspuns corect c) 2**

Indicații: corecte sunt variantele 3 și 4

2. **Răspuns corect e) 2 14**

3. **Răspuns corect e) 112**

4. **Răspuns corect b) 2 4 6**

5. **Răspuns corect a) 4**

6 3

6. **Răspuns corect c) neenUB**

7. **Răspuns corect d) 1225**

Indicații: $49+48+\dots+1=50*49/2=1225$ (suma Gauss)

8. **Răspuns corect b) dcafe;
dcbaf**

9. **Răspuns corect f) 350**

Indicații: meniu= felul întâi + felul doi sau

meniu= felul întâi + felul doi + desert

10. **Răspuns corect c) 31**

Indicații: Graful dat are 14 muchii. K_{10} are 45 muchii. $45-14=31$ muchii trebuie adăugate pentru a obține un K_{10}

11. **Răspuns corect a) 0**

12. **Răspuns corect f) 30**

Indicații: Într-un arbore binar se face diferența între fiul stâng și fiul drept. Sunt 5 configurații posibile * 6 etichetări diferite=30 cazuri

13. **Răspuns corect b) 5**

Indicații: descendenții nodului 4 sunt: **3, 6, 11, 13, 15**

14. **Răspuns corect d) $5^2 \cdot 2^{276}$**

Indicații: Fie n numărul de noduri din graf. Un nod este izolat și cu restul se construiesc grafuri neorientate cu $n-1$ noduri ($2^{(n-1)} \cdot 2^{(n-2)/2}$ cazuri). Orice nod poate fi ales ca nod izolat. Total grafuri neorientate $n \cdot 2^{(n-1)} \cdot 2^{(n-2)/2}$.

15. **Răspuns corect c) $p \hat{a}^{tm} n^3$**

Variantă 14

Indicații și răspunsuri

1. **Răspuns corect c) 3**
2. **Răspuns corect d) 3 1**
3. **Răspuns corect a) 5 30
5 70**
4. **Răspuns corect b) 2 600**
5. **Răspuns corect c) 8 11 14**
6. **Răspuns corect e) EBPŪ-UPB**
7. **Răspuns corect d) 4 950**

Indicații: $(n-1)+(n-2)+\dots+1=n*(n-1)/2$ (suma Gauss), unde n este numărul de elemente al tabloului unidimensional

8. **Răspuns corect b) 1 3122**
9. **Răspuns corect a) 1 212 ; 4 322**
10. **Răspuns corect b) 15**
11. **Răspuns corect c) 50**
12. **Răspuns corect d) 50**

Indicații: Pentru ca arborele binar să aibă înălțime minimă se ocupă fiecare nivel x cu câte 2^x noduri, unde x este de la 0 la 5 (rădăcina este pe nivelul 0). Pe nivelul 5 avem 32 frunze. S-au folosit $2^6 - 1 = 63$ noduri. Pe nivelul 6 se adaugă încă 37 de noduri. Astfel, deduc că rămân 13 frunze pe nivelul 5, deoarece 18 noduri au câte doi fii și un nod are un fiu ($32 - 18 - 1 = 13$ frunze), iar pe nivelul 6 sunt 37 frunze. Total: 50 frunze.

13. **Răspuns corect d) 5**
Indicații: $L_1: 10, 1, 5, 7$; $L_2: 10, 1, 5, 12$; $L_3: 10, 4, 3, 11$;
 $L_4: 10, 4, 3, 13$; $L_5: 10, 4, 3, 15$;
14. **Răspuns corect d) $5^{2^{52}} 2^{552}$**

Indicații: Fie n numărul de noduri din graf. Un nod este izolat și cu restul se construiesc grafuri orientate cu $n-1$ noduri ($4^{(n-1) \wedge (n-2)/2}$ cazuri = $2^{(n-1) \wedge (n-2)}$ cazuri). Orice nod poate fi ales ca nod izolat. Total grafuri orientate $n \cdot 2^{(n-1) \wedge (n-2)}$

15. **Răspuns corect a) $O(n)$**

Variantă 15

Indicații și răspunsuri

1. Răspuns corect f) 90
2. Răspuns corect b) n și $i > 0$
3. Răspuns corect a)
(Limbajul C/C++)
`if (a > b && a % 2 == 0 || b >= a && b % 2 == 0) c = a;
else c = b;`
(Limbajul Pascal)
`if ((a > b) and (a mod 2 = 0)) or ((b >= a) and (b mod 2 = 0))
then c := a
else c := b;`
4. Răspuns corect e) 8
5. Răspuns corect c) (5, 8, 4, 0, 4, 5, 3, 6, 7, 8)
6. Răspuns corect a)
(Limbajul C/C++)
`i < j && i + j < n + 1`
(Limbajul Pascal)
`(i < j) and (i + j < n + 1)`
7. Răspuns corect f) 2
8. Răspuns corect e) 3
9. Răspuns corect d) 24
10. Răspuns corect a) 2
11. Răspuns corect c) 2b4d4
12. Răspuns corect b) diarrafetbdul
13. Răspuns corect f) -8
14. Răspuns corect b) 2 0
15. Răspuns corect a) 9

Variantă 16

Indicații și răspunsuri

1. Răspuns corect d) 31
2. Răspuns corect a)
(Limbajul C/C++)
`n și i=i-1;`
(Limbajul Pascal)
`n și i:=i-1;`
3. Răspuns corect a)
(Limbajul C/C++)
`if (a>b && a%2==0 && b%2==0) c=a;`
`if (a>b && a%2==0 && b%2!=0) c=b;`
(Limbajul Pascal)
`if (a>b) and (a mod 2=0) and (b mod 2=0) then c:=a;`
`if (a>b) and (a mod 2=0) and (b mod 2<>0) then c:=b;`
4. Răspuns corect f) 44
5. Răspuns corect a) 1, 2, 6, 8, 10
6. Răspuns corect
(Limbajul C/C++) c) `i+j==n+2`
(Limbajul Pascal) c) `i+j=n+2`
7. Răspuns corect a) $x \in (-\infty, -10) \cup [10, 100)$
8. Răspuns corect e) 2
9. Răspuns corect b) 23
10. Răspuns corect f) 3
11. Răspuns corect a) 2c3d4
12. Răspuns corect c) eAiuuueUIa
13. Răspuns corect c) 7 8 9 2
14. Răspuns corect d) 2 0
15. Răspuns corect e) 6284

Varianta 17

Indicații și răspunsuri

1. **Răspuns corect b) 2**
2. **Răspuns corect d) 4**
3. **Răspuns corect c) 90**
4. **Răspuns corect a) 1**
5. **Răspuns corect f) 1110**
Indicații: numărul ciclomatic $=m-n+p$
6. **Răspuns corect e) o1i 2020**
7. **Răspuns corect d) Bucuresti 2020 ADMIS**
8. **Răspuns corect a) 864**
9. **Răspuns corect f) 4**
10. **Răspuns corect f) nici o valoare**
11. **Răspuns corect a) $\frac{3}{4} \cdot (n + 1) \cdot n$**
12. **Răspuns corect c) int/int/integer**
13. **Răspuns corect f) 2014**
14. **Răspuns corect c) $O(2^n)$**
15. **Răspuns corect a) $O(n \cdot \log(n))$**

Varianta 18

Indicații și răspunsuri

1. Răspuns corect b) 2

2. Răspuns corect d) 4

3. Răspuns corect a) 0.83

4. Răspuns corect a) 1

5. Răspuns corect f) 6

Indicații: $2^{15} = 2^{n(n-1)/2} = 32768$

6. Răspuns corect f) 0202 iloP

7. Răspuns corect d) Politehnica Bucuresti 2020XXXXXXXXXXXX

8. Răspuns corect f) 16460640

9. Răspuns corect c) $O(n)$

Indicații: Algoritmul de interclasare

10. Răspuns corect c) $\Theta(2^n)$

Indicații: Turnurile din Hanoi.

11. Răspuns corect a) $\frac{5}{4} \cdot (n+1) \cdot n$

12. Răspuns corect c) double/double/real

13. Răspuns corect e) 34340 1

14. Răspuns corect a) $O(n+m)$

15. Răspuns corect d) $O(n^2)$

Varianta 19

Indicații și răspunsuri

1. **Răspuns corect:** b) 45
2. **Răspuns corect:** e) 2 instrucțiuni
Indicații: pentru $n \geq 1$ se execută doar cele două atribuiri.
3. **Răspuns corect:** f) 4 componente
Indicații: $\{1\}, \{2,3,4,5\}, \{6\}, \{7\}$
4. **Răspuns corect:** d) 6
5. **Răspuns corect:** c)
Limbajul C++: `int &x, int y;` **Limbajul C:** `int *x, int y;`
Limbajul Pascal: `var x:integer; y:integer;`
6. **Răspuns corect:** b) 9^{138}
Indicații: Sunt $3^{n(n-1)/2}$ grafuri complete orientate. Pentru $n=24$ avem 3^{276}
7. **Răspuns corect:** a) 20v
8. **Răspuns corect:** f) 1
Indicații: Se șterge de exemplu muchia $(1, 2)$.
9. **Răspuns corect:** e) 16
10. **Răspuns corect:** a) $O(m \cdot \log(n))$
Indicații: Avem o structură repetitivă cu valori de la 1 la m , în interiorul căreia avem o căutare binară - deci $O(m \cdot \log(n))$
11. **Răspuns corect:** c) $O(n)$
Indicații: Citirea vectorului are complexitatea $O(n)$, subprogramul are complexitatea $O(\log(n))$. Prin urmare ordinul de complexitate al secvenței este $O(n)$.
12. **Răspuns corect:** b) $4^{19}-1$
Indicații: Sunt $1+2+2^2+2^3+\dots+2^{37}$ noduri, adică $2^{38}-1=4^{19}-1$
13. **Răspuns corect:** a) 42 72 152 1518
14. **Răspuns corect:** b) 1001997
Indicații: Fie $v=[n, n-1, \dots, 2, 1]$ unde n se află pe poziția 1, $n-1$ pe poziția 2, 1 pe poziția n . Pentru fiecare element de pe poziția i , $i \leq n/2$ se execută $3+2(i-1)$ pași. Sunt în total $3(n-1) + 2[(n(n-1))/2]$ pași, adică $(n-1)(n+3)$ pași. Pentru $n=6$ avem $5(6+3)=45$ pași, iar pentru $n=1000$ se execută $999(1000+3)=999 \times 1003=1001997$ pași.

15. **Răspuns corect:** a) **15**

Indicații: valorile afișate vor fi: 1 1 1 2 1 2 1 1 2 1 3 1 1 2 1 deci în total instrucțiunea de decizie se execută de 15 ori.

Varianta 20

Indicații și răspunsuri

1. **Răspuns corect:** b) 2 apeluri

Indicații: $f(720, 2)$ și $f(120, 1)$.

2. **Răspuns corect:** b) **c, a**

Indicații: În stivă se rețin valorile variabilelor locale (variabila **a**) și valorile parametrilor transmiși prin valoare (variabila **c**).

3. **Răspuns corect:** c) $O(n^2)$

4. **Răspuns corect:** b) **Limbajul C/C++: $!(x*y+y-3*x-3 \geq 0)$**

Limbajul Pascal: $NOT(x*y+y-3*x-3 >= 0)$

Indicații: $x > -1$ și $y < 3$ implică $(x+1)(y-3) < 0$ echivalent cu $xy+y-3x-3 < 0$ echivalent cu $!(x*y+y-3*x-3 \geq 0)$ în limbajul C++, respectiv $NOT(x*y+y-3*x-3 \geq 0)$ în limbajul Pascal.

5. **Răspuns corect:** a) 3

Indicații: $\{0, 1, 2, 3, 4\}, \{0, 1, 4, 5\}, \{0, 2, 3, 5\}$

6. **Răspuns corect:** b) **aib**

Indicații: Soluțiile care au două vocale în ordinea generării sunt: abe, abi, abu, ace, aci, acu, aeb, aec, **aib**, aic, aie

7. **Răspuns corect:** c) 7

Indicații: $2+2+2+2+2+2; 2+2+2+3+3; 2+2+3+5; 2+3+7; 2+5+5; 3+3+3+3; 5+7$

8. **Răspuns corect:** a) **{biologie, mate, info};**

Indicații: Soluțiile în ordinea generării sunt: {fizica, biologie, chimie}; {fizica, biologie, mate}; {fizica, biologie, info}; {fizica, chimie, mate}; {fizica, chimie, info}; {fizica, mate, info}; {biologie, chimie, mate}; {biologie, chimie, info}; **{biologie, mate, info}**; {chimie, mate, info}

9. **Răspuns corect:** c) 2^k

10. **Răspuns corect:** a) **linia 5, coloana 7**

Indicații: Elementul de pe linia i , coloana j are valoarea $(i-1)m+j$. $123=3m+3$ deci $m=40$. Prin urmare sunt 40 elemente pe o linie $167=40*4+7$. Deci $i=5, j=7$.

11. **Răspuns corect:** e) **n**

Indicații: Cazul cel mai defavorabil este atunci când n este număr prim. În acest caz, în afara structurii repetitive sunt 2 instrucțiuni de atribuire iar în cadrul structurii repetitive sunt $n-2$ instrucțiuni de incrementare. În total n instrucțiuni.

12. **Răspuns corect:** e) 2 4 6

13. **Răspuns corect:** b) 4 4 3

Indicații: Subprogramul calculează recursiv $x_1^n + x_2^n$ unde $s = x_1 + x_2$ iar $p = x_1 * x_2$.

Prin urmare $x_1^4 + x_2^4 = 82$ dacă $x_1 + x_2 = 4$ și $x_1 * x_2 = 3$, atunci $x_1 = 3$ și $x_2 = 1$ deci $3^4 + 1 = 82$.

14. **Răspuns corect:** b) $n=6$; $k=1$

Indicații: Se construiește un tablou bidimensional care are elemente cu valoarea 1 pe primele k diagonale paralele cu cele două diagonale, în rest, elemente cu valoarea 2.

15. **Răspuns corect:** c) $\alpha=j$; $\beta= 4-j-k$

Variantă 21

Indicații și răspunsuri

1. **Răspuns corect d)**

2. **Răspuns corect d)**

3. **Răspuns corect c)**

Indicație: Atât în **Limbajul Pascal** cât și în **Limbajul C/C++**, ; este instrucțiunea vidă.

4. **Răspuns corect d)**

5. **Răspuns corect c)**

6. **Răspuns corect c)**

7. **Răspuns corect d)**

8. **Răspuns corect c)**

9. **Răspuns corect d)**

10. **Răspuns corect b)**

11. **Răspuns corect d)**

12. **Răspuns corect d)**

Indicație: Funcția va returna 0 dacă toate elementele vectorului sunt în ordine descrescătoare. În evaluarea expresiei $v[n-1] < v[n] \parallel f(n-1)$ are loc o scurtcircuitare, dacă o singură dată relația $v[n-1] < v[n]$ este adevărată funcția va returna valoarea 1 (true) și se întrerupe apelul recursiv.

13. **Răspuns corect d)**

Indicație: Numărul grafurilor parțiale este egal cu numărul submulțimilor mulțimii muchiilor adică 2^4 .

14. **Răspuns corect a)**

Indicație: formula este $\sum_{i=1}^{10} \sum_{j=1}^i \sum_{k=1}^j 1 = \sum_{i=1}^{10} \sum_{j=1}^i j = \sum_{i=1}^{10} (1 + 2 + \dots + i) =$

$$\sum_{i=1}^{10} \frac{i(i+1)}{2} = \frac{1}{2} \sum_{i=1}^{10} (i^2 + i) = 220$$

15. **Răspuns corect d)**

Varianta 22

Indicații și răspunsuri

1. **Răspuns corect c)**

Indicație: Numărul de frunze este egal cu numărul de factori primi din descompunerea numărului.

2. **Răspuns corect b)**

3. **Răspuns corect c)**

4. **Răspuns corect d)**

5. **Răspuns corect b)**

6. **Răspuns corect a)**

7. **Răspuns corect d)**

Indicație:

a) Deoarece nodul 7 are gradul 6, el este adiacent cu toate celelalte noduri, deci nu este posibil ca un nod să aibă gradul zero.

b) Deoarece nodul 7 are gradul 6, el este adiacent cu toate celelalte noduri. Nodul 6 are gradul 5, aceasta înseamnă că el este adiacent cu 7 și încă 4 noduri, deci trebuie să existe 4 noduri care să aibă minim gradul 2 și nu există.

c) Deoarece nodul 7 are gradul 6, el este adiacent cu toate celelalte noduri. Pentru ca nodul 5 să poată avea gradul 2 trebuie ca încă un nod să aibă minim gradul 2.

e) Deoarece nodul 7 are gradul 5, el este adiacent cu 5 noduri și nu pot exista două noduri care să aibă gradul 0.

f) Deoarece nodul 7 are gradul 3, el este adiacent cu 3 noduri. Deoarece nodul 6 are gradul 2, el este adiacent cu 7 și cu încă un nod. Dar nu mai există un nod de grad 2.

8. **Răspuns corect d)**

9. **Răspuns corect b)**

10. **Răspuns corect d)**

11. **Răspuns corect a)**

12. **Răspuns corect d)**

13. **Răspuns corect b)**

Indicație:

Pentru $n=3$ avem un singur ciclu hamiltonian 1 2 3 1.

Pentru $n=4$, îl intercalăm pe 4 în toate modurile posibile și obținem ciclurile hamiltoniene: 1 4 2 3 1, 1 2 4 3 1, 1 2 3 4 1.

Presupunem ca în graful cu n noduri avem $\frac{(n-1)!}{2}$ cicluri hamiltoniene.

În graful cu $n+1$ noduri, intercalând pe $n+1$ în toate modurile posibile obținem $n \cdot \frac{(n-1)!}{2} = \frac{n!}{2}$ cicluri hamiltoniene.

14. **Răspuns corect d)**

Indicație: Funcția va returna 0 dacă toate elementele vectorului sunt în ordine descrescătoare. În evaluarea expresiei $v[n-1] < v[n] \parallel f(n-1)$ are loc o scurtcircuitare, dacă o singură dată relația $v[n-1] < v[n]$ este adevărată funcția va returna valoarea 1 (true) și se întrerupe apelul recursiv.

15. **Răspuns corect d)**

Varianta 23

Indicații și răspunsuri

1. **Răspuns corect:** c) $b+a/10!=b\%c*a/c$ (limbaj C++/C)
 $b+a \text{ div } 10 <> b \text{ mod } c * a \text{ div } c$ (limbaj Pascal)
2. **Răspuns corect:** f) 11
Indicații: suma gradelor trebuie să fie un număr par. Pentru $n=13$ și $d=11$ suma gradelor ar fi 143.
3. **Răspuns corect:** c) 7
Indicații: $i=1$; $i=2$; $i=4$; $i=8$; $i=16$; $i=32$; $i=64$
4. **Răspuns corect:** e) 8
Indicații: se elimină cifrele, dar nu cele care sunt precedate de o cifră ștersă
5. **Răspuns corect:** b) $n*(n-1)/2$
Indicații: se execută $(n-1) + (n-2) + \dots + 2 + 1$ comparații
6. **Răspuns corect:** c) -4
Indicații: $x=15$; $x=14$; $x=7$; $x=2$; $x=1$; $x=-4$;
7. **Răspuns corect:** d) 13
Indicații: Se generează: 1003, 1012, 1021, 1030, 1102, 1111, 1120, 1201, 1210, 1300, 2002, 2011, 2020
8. **Răspuns corect:** a) suma elementelor de sub diagonala principală exclusiv elementele diagonalei principale
9. **Răspuns corect:** e) $h.g.c[2]$
10. **Răspuns corect:** e) 1, 4, 5, 6, 8
Indicații: lanțurile elementare de lungime 3 sunt: (1,2,3,4), (1,2,3,6) (1,2,7,5) (8,2,3,4) (8,2,3,6) (8,2,7,5) (3,2,7,5) (6,3,2,7) (4,3,2,7)
 Nodurile 1, 4, 5, 6 și 8 apar în câte 3 lanțuri elementare, celelalte noduri apar de mai multe ori.
11. **Răspuns corect:** f) 6 3 1 3 2 1
Indicații: Primul "for" atribuie tabloului: 0 5 1 3 2 1
 $a[a[6]]=2*6\%7$ $a[1]=5$
 $a[a[5]]=2*5\%7$ $a[2]=3$
 $a[a[4]]=2*4\%7$ $a[3]=1$
 $a[a[3]]=2*3\%7$ $a[1]=6$
12. **Răspuns corect:** c) $s(2020, 2)=4$ și reprezintă suma exponenților divizorilor primi din descompunerea în factori primi a numărului 2020
13. **Răspuns corect:** d) 1792
Indicații: Sunt $2^6=64$ grafuri cu 4 noduri (nodurile 3, 4, 5, 6).
 Dacă există muchia [1, 2] atunci 2 se conectează cu 3 sau 4 sau 5 sau 6. Deci în acest caz avem 4 variante.
 Dacă 1 se conectează cu 3 sau 4 sau 5 sau 6 atunci și 2 se conectează cu două dintre nodurile 3, 4, 5, 6. În acest caz avem $4*6=24$ variante.
 În total avem $4+24=28$ variante de conectare pentru 1 și 2. În total avem $28*64=1792$ grafuri cu proprietatea cerută.
14. **Răspuns corect:** f) 3 11 2 2 10 2 1 9 2 1 6

15. **Răspuns corect:** f) Aplicând metoda de sortare prin inserție se poate obține ca etapă intermediară tabloul $v = (1, 3, 4, 2, 5, 7, 6)$
Indicații: Dacă se parcurge tabloul de la primul element la ultimul, prin inserție avem: $(2, 3, 4, 5, 1, 7, 6)$ $(1, 2, 3, 4, 5, 7, 6)$ $(1, 2, 3, 4, 5, 6, 7)$
Bubble Sort:
prima traversare: se schimbă **4** cu **2**, **5** cu **1**, **7** cu **6** și se obține tabloul $(3, 2, 4, 1, 5, 6, 7)$; a doua traversare: se schimbă **3** cu **2**, **4** cu **1** și se obține tabloul $(2, 3, 1, 4, 5, 7)$; a treia traversare: se schimbă **3** cu **1** și se obține tabloul $(2, 1, 3, 4, 5, 6, 7)$; a patra traversare: se schimbă **2** cu **1** și se obține tabloul sortat
- b) interclasare: dacă se interclasează $(2, 3, 4, 5)$ cu $(1, 6, 7)$, se compară **1** cu **2** și **1** devine primul în tablou, apoi **2** cu **6**, **3** cu **6** etc.
Dacă se interclasează $(2, 3, 4)$ cu $(1, 5, 6, 7)$ se compară **1** cu **2** și **1** devine primul în tablou, apoi **2** cu **5**, **3** cu **5** etc.
- c) prin selecția minimului/maximului se fac cel mult $n-1$ interschimbări
- d) la prima parcurgere se compară **3** cu **2** și minimul devine **2**. Tabloul devine $(1, 4, 2, 5, 3, 7, 6)$. La a doua parcurgere minimul inițial este **4**, apoi devine **2** și se compară cu **3**, iar **2** este plasat pe a doua poziție a tabloului.
- e) După prima traversare se obține tabloul $(3, 2, 4, 1, 5, 6, 7)$

Varianta 24

Indicații și răspunsuri

1. **Răspuns corect:** d) număr natural impar de o singură cifră
Indicații: $n \% 2 == 1$ este adevărată pentru numere naturale impare
2. **Răspuns corect:** f) $i + j = n + 1$
3. **Răspuns corect:** d) este hamiltonian dar nu eulerian
Indicații: fiecare nod are gradul 9, care nu este un număr par
4. **Răspuns corect:** c) $a[i] - a[i-1] != d$ (C++/C) respectiv $a[i] - a[i-1] <> d$ (Pascal)
5. **Răspuns corect:** b) 2
Indicații: cele mai lungi lanțuri elementare sunt (5, 7, 2, 3, 6), (5, 7, 2, 3, 4) și au în mijloc nodul 2
6. **Răspuns corect:** a) 1326
Indicații: se compară la fiecare pas ultima cifră a numărului **a** cu ultima cifră a numărului **b** și cea mai mică dintre acestea este adăugată numărului **p**.
 $a = 11357$
 $b = 1426$
7. **Răspuns corect:** c) 19
Indicații: se determină maxim pentru toți indicii, mai puțin pentru ultimul indice
8. **Răspuns corect:** c) orientat cu 4 noduri și 6 arce
Indicații: Matricea are 4 linii și 4 coloane, deci graful are 4 noduri. Matricea nu este simetrică, deci nu este graf neorientat. Are 6 elemente nenule, deci are 6 arce.
9. **Răspuns corect:** e) **noram** și **nramo**
Indicații: după ordonarea alfabetică a literelor cuvântului **roman** se obține **amnor**. Dacă nu le ordonează și consideră ca primă soluție cuvântul dat se obține răspunsul **a**
10. **Răspuns corect:** c) o rama alba
Indicații: Se elimină spațiile din șir și se verifică dacă este palindrom. Singurul care nu este palindrom este **oramaalba**
11. **Răspuns corect:** d) 14
Indicații: $f(3) = f(2) + 2 * f(0) = 3 + 2 * 1 = 5$
 $f(2) = f(1) + 2 * f(-1) = 1 + 2 = 3$ deci $f(3)$ are 4 apeluri
 $f(5) = f(4) + 2 * f(2)$ $f(2) = f(1) + 2 * f(-1)$
 $f(4) = f(3) + 2 * f(1)$ $f(3) = f(2) + 2 * f(0)$ $f(2) = f(1) + 2 * f(-1)$
 $f(5)$ și $f(3)$ nu se numără pentru că sunt apeluri din programul principal. Se cere numărul de autoapeluri!
12. **Răspuns corect:** c) 14
Indicații: se generează tabloul
1 2 3 4
5 6 7 3
8 9 6 2
10 8 5 1
13. **Răspuns corect:** f) 112 și 166
Indicații: $f(95) = f(1 + f(97)) = f(110) = 112$;
 $f(97) = f(1 + f(99)) = f(107) = 109$;

$$f(99) = f(1 + f(101)) = f(1 + 103) = f(104) = 106;$$

Se observă că plecând de la $x=99$, dacă x scade cu 2, $f(x)$ crește cu 3.

Plecând de la 99 avem $99-59=40$; $40/2=20$; $20*3=60$; $60+106=166$ sau

Plecând de la 95 avem $95-59=36$; $36/2=18$; $18*3=54$; $54+112=166$

14. **Răspuns corect:** e) Cel mai mic număr de operații s-a efectuat pentru z .

Indicații: ca operații avem:

operații comune la toate tablourile:

-numărul de comparații pentru determinarea minimului este 6

-inițializarea indicelui valorii minime 3 operații

-verificare dacă minimul se află pe poziția i (ca să nu fac interschimbare cu el însuși)

3 operații

diferențe:

-(d1) nr de interschimbări

-(d2) nr de actualizări ale indicelui minimului

v: $d1=3$ și $d2=3$

x: $d1=2$ și $d2=2$

y: $d1=2$ și $d2=2$

z: $d1=1$ și $d2=2$

15. **Răspuns corect:** d) divizorii primi ai lui x și numărul tuturor divizorilor lui x

EXEMPLU: pentru $x=36$ se afișează 2 3 9

Indicații: este o descompunere în factori primi care afișează divizorii primi (la prima apariție în descompunere) și returnează numărul total de divizori calculat ca produs de exponenți plus 1.

$$36=2^2 \cdot 3^2$$

numărul divizorilor este $(2+1) \cdot (2+1) = 9$

Varianta 25

Indicații și răspunsuri

1. **Răspuns corect:** b) $4 \cdot a \cdot (a-1) < a \cdot a - 2$

Indicații: $4 \cdot a \cdot (a-1) < a \cdot a - 2$ se scrie astfel $3a^2 - 4a + 2 < 0$ unde delta este -8 , rezultă că ecuația de gradul al II-lea va fi pozitivă mereu.

2. **Răspuns corect:** a) 167238945

Indicații:

			Se afișează
	i=4	j=0	A[4][0]=1
j<>4 (A)	i=3	j=0	A[3][0]=6
	i=3	j=1	A[3][1]=7
j<>4 (A)	i=2	j=1	A[2][1]=2
	i=2	j=2	A[2][2]=3
j<>4 (A)	i=1	j=2	A[1][2]=8
	i=1	j=3	A[1][3]=9
j<>4 (A)	i=0	j=3	A[0][3]=4
	i=0	j=4	A[0][4]=5
j<>4 (F)			

3. **Răspuns corect:** e) 7

Indicații:

Ad (3)

3		
---	--	--

E1 ()

9		
---	--	--

E1 ()

--	--	--

Ad (5)

9	5	
---	---	--

Ad (7)

7		
---	--	--

Ad (2)

9	5	2
---	---	---

Ad (9)

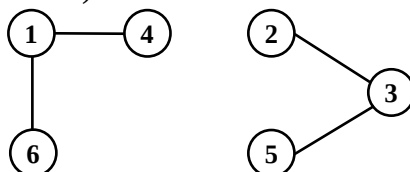
7	9	
---	---	--

E1 ()

5	2	
---	---	--

4. **Răspuns corect:** c) 2

Indicații:

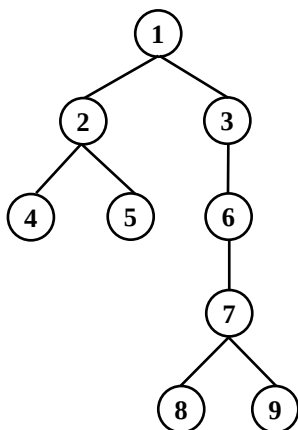


Componenta conexă I este formată din nodurile: 1, 4, 6

Componenta conexă II este formată din nodurile: 2, 3, 5

5. **Răspuns corect:** f) 9 4

Indicații:



6. **Răspuns corect:** e) 8

Indicații: 114, 123, 132, 141, 213, 222, 231, 312

7. **Răspuns corect:** a) **xxmmnn**

Indicații: Instrucțiunea repetitivă **for** parcurge șirul "examen" până la **penultimul caracter** inclusiv și verifică dacă valoarea elementului de pe poziția **i** este strict mai mică din punct de vedere lexicografic decât valoarea elementului următor, **a[i+1]**. Dacă acest element este strict mai mic decât următorul, lui **a[i]** i se atribuie valoarea lui **a[i+1]**.

8. **Răspuns corect:** b) **4324**

97

Indicații: Deoarece există o instrucțiune **if** cu condiția **n<=100**, înseamnă că funcția **numar** se va opri când **n=82**. Până atunci va verifica fiecare cifră a numărului **n**. Dacă cifra este mai mică decât 5 aceasta va fi afișată imediat deoarece afișarea se face înainte de reapelarea funcției (4, 3, 2 și 4). Dacă cifra este mai mare decât 5 (9 și 7) aceasta va fi afișată la sfârșit, în ordine inversă a găsirii ei, deoarece afișarea se execută după reapelarea funcției.

9. **Răspuns corect:** b) 3

Indicații: Liniile de cod află câți divizori primi are valoarea memorată în variabila **a**. Variabila **c**, care inițial are valoarea 2, este incrementată cu 1 până când valoarea variabilei **a** devine 1. Valoarea variabilei **a** ajunge la 1 deoarece este împărțită, pe rând, dacă este posibil, la valorile luate de variabila **c**. De câte ori se găsește un număr în variabila **c** care divide valoarea variabilei **a**, valoarea variabilei **b** crește cu 1.

10. **Răspuns corect:** c) **v=[5, 8, 2, 6, 6, 5, 4, 4]**

Indicații:

i=0	v[0]<5 (F)	v[0]=5	v[7]>v[0] (A)	v[7]=4
i=1	v[1]<5 (F)	v[1]=8	v[6]>v[1] (F)	v[6]=4
i=2	v[2]<5 (A)	v[2]=2	v[5]>v[2] (A)	v[5]=5
i=3	v[3]<5 (A)	v[3]=6	v[4]>v[3] (F)	v[4]=6

11. **Răspuns corect:** d) $\frac{c \cdot (c+2)}{4}$

Indicații: $2 + 4 + 6 + \dots + n = 2 * \left(1 + 2 + 3 + \dots + \frac{n}{2}\right) = 2 * \frac{\frac{n}{2} * (\frac{n}{2} + 1)}{2}$.

Instrucțiunea repetitivă for calculează suma $1 + 2 + 3 + \dots + \frac{n}{2}$.

12. **Răspuns corect:** a) 1 2 3 4 5 6 7

Indicații:

i=1	a[1]=7	a[7]=1
i=2	a[2]=6	a[6]=2
i=3	a[3]=5	a[5]=3

i=4	a[4]=4	a[4]=4
i=5	a[5]=3	a[3]=5
i=6	a[6]=2	a[2]=6
i=7	a[7]=1	a[1]=7

13. **Răspuns corect:** d) 101

Indicații: Instrucțiunile respective numără câte cifre impare există, în total, în intervalul [200,300]. Prin urmare pe poziția cifrei sutelor, o cifră impară va apărea doar pentru numărul 300.

Pe poziția zecilor, o cifră impară, se va regăsi de exact 50 de ori în acest interval, pentru numerele de forma $\overline{2ab}$ unde $a \in \{1, 3, 5, 7, 9\}$ iar $b \in \{0, 1, 2, \dots, 9\}$.

Cifra unităților va avea o valoare impară pentru numere de forma $\overline{2ab}$ unde $a \in \{0, 1, 2, \dots, 9\}$ iar $b \in \{1, 3, 5, 7, 9\}$. Putem deduce de aici că în intervalul [200,300] pe poziția cifrei unităților vom regăsi o cifră impară de exact 50 de ori.

Pe poziția sutelor există o singură cifră impară. Deci $50 + 50 + 1 = 101$.

14. **Răspuns corect:** c) determinantul matricei

Indicații: Subprogramul **mat** calculează recursiv determinantul matricei **d**, primită ca parametru, prin descompunerea acesteia după linii și coloane.

În variabila **e**, declarată tot ca tablou bidimensional, este reținută matricea rezultată după descompunerea matricei inițiale, în funcție de elementul **d[i][j]**. Prin urmare, tabloul bidimensional **e** va reține o matrice de **n-1** linii și **n-1** coloane.

Aceasta metodă se aplică recursiv până când matricea reținută în tabloul bidimensional **e** va avea un singur element.

15. **Răspuns corect:** b) 5417032963258410

Indicații: În cazul de față funcția recursivă denumită **afis** se va opri în momentul în care **k** devine 1. De menționat că, în cazul de față, **k** pornește de la valoarea 8 care este valoarea sumei indicilor lui **a[4][4]**, elementul din dreapta jos.

Valoarea variabilei **k** scade cu 1 la fiecare reapelare a funcției. Pentru **k** par se vor afișa elementele de la stânga spre dreapta iar pentru **k** impar se vor afișa elementele de la dreapta spre stânga.

$$v = \begin{pmatrix} v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \\ v_{31} & v_{32} & v_{33} & v_{34} \\ v_{41} & v_{42} & v_{43} & v_{44} \end{pmatrix}$$

$$V = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 0 & 1 \\ 2 & 3 & 4 & 5 \end{pmatrix}$$

Varianta 26

Indicații și răspunsuri

1. **Răspuns corect:** e) C++: $(a \% 3 + a \% 7) / 9$

Pascal: $(a \text{ MOD } 3 + a \text{ MOD } 7) \text{ DIV } 9$

Indicații: Restul împărțirii unui număr la 3 poate fi 0, 1 sau 2. Restul împărțirii unui număr la 7 poate fi maxim 6. În concluzie, suma acestor două resturi nu poate fi mai mare de 8.

2. **Răspuns corect:** b) $v = (5, 3, 4, 8, 6, 2, 1, 9)$

Indicații: Instrucțiunile din cadrul instrucțiunii repetitive **while**

(C++: $v[i] = v[i] + v[j]; v[j] = v[i] - v[j]; v[i] = v[i] - v[j];$ sau

Pascal: $v[i] := v[i] + v[j]; v[j] := v[i] - v[j]; v[i] := v[i] - v[j];$)

interschimbă valoarea aflată în $v[i]$ cu valoarea aflată în $v[j]$.

i=0	j=1	v[0]=5	v[1]=3
i=2	j=3	v[2]=4	v[3]=8
i=4	j=5	v[4]=6	v[5]=2
i=6	j=7	v[6]=1	v[7]=9

3. **Răspuns corect:** f) **poLiTEHnica**

Indicații: Toate literele mai mici decât litera n, în ordine alfabetică, până la poziția 7 din șirul de caractere memorat în variabila a, sunt transformate în litere mari (spre exemplu: l devine L)

4. **Răspuns corect:** d) $A = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 2 & 4 \\ 2 & 2 & 4 & 3 \\ 3 & 4 & 3 & 6 \end{pmatrix}$

Indicații: Cele două instrucțiuni repetitive **while** parcurg toate elementele tabloului bidimensional **A**. Elementele care respectă condiția instrucțiunii **if** ($i+j$ să fie par) sunt: $A[0][0]=0$, $A[0][2]=2$, $A[1][1]=2$, $A[1][3]=4$, $A[2][0]=2$, $A[2][2]=4$, $A[3][1]=4$ și $A[3][3]=6$.

Celelalte elemente au valoarea celui mai mare dintre indicii i și j : $A[0][1]=1$, $A[0][3]=3$, $A[1][0]=1$, $A[1][2]=2$, $A[2][1]=2$, $A[2][3]=3$, $A[3][0]=3$ și $A[3][2]=3$.

5. **Răspuns corect:** d) 11

Indicații:

Ad (3)

Ad (7)

Ad (5)

El ()

El ()

Ad (8)

3

7
3

5
7
3

7
3

3

8
3

6. **Răspuns corect:** d) 24

Indicații: 4a8b (4185, 4189, 4581, 4589, 4981, 4985) - 6 numere

8a4b - 6 numere

a4b8 - 6 numere

a8b4 - 6 numere

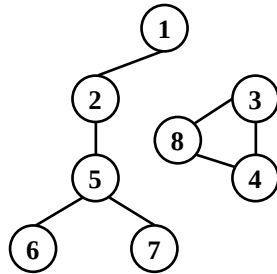
7. **Răspuns corect:** f) 2 2 2 1 1 2

Indicații: Nodul cu numărul 2 din varianta de răspuns **a** indică faptul că este adiacent cu 5 noduri chiar dacă gradul nodului 3 este 0. Același raționament se aplică și pentru varianta **d**. Într-un graf neorientat suma gradelor tuturor nodurilor trebuie să fie un număr par. Prin urmare variantele **b** și **e** sunt false. Nodul cu numărul 3 din varianta de răspuns c are gradul 7 chiar dacă graful are 6 noduri.

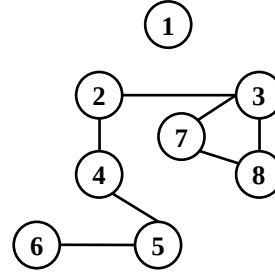
8. **Răspuns corect:** d) $T = (0 \ 5 \ 7 \ 3 \ 1 \ 3 \ 1 \ 2)$

Indicații:

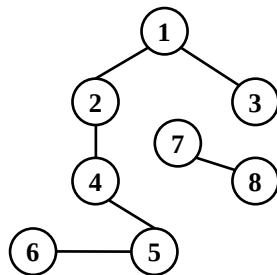
Reprezentarea grafică a variantei **a**:



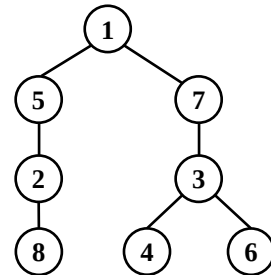
Reprezentarea grafică a variantei **b**:



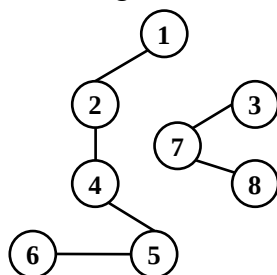
Reprezentarea grafică a variantei **c**:



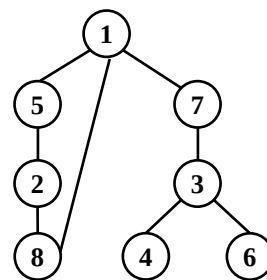
Reprezentarea grafică a variantei **d**:



Reprezentarea grafică a variantei **e**:



Reprezentarea grafică a variantei **f**:



9. **Răspuns corect:** b) a^2+1

Indicații: Cele două instrucțiuni repetitive **while** și **do... while** / **repeat... until** aflate în interiorul instrucțiunii **for** parcurg intervalul $[1, a]$. Instrucțiunea **while** parcurge intervalul $[1, i]$ iar instrucțiunea **do... while** / **repeat... until** parcurge intervalul complementar $[i+1, a]$.

Deoarece aceste instrucțiuni se află într-un **for** care parcurge același interval $[1, a]$, variabila **s** va avea valoarea a^2 . Va afișa a^2+1 deoarece instrucțiunea **do... while**

/ **repeat... until** va incrementa cu **1** variabila **s** în ultima apelare a instrucțiunii chiar dacă condiția este falsă deoarece este instrucțiune repetitivă cu test final.
Pentru C++ nu se va afișa **a²**, ci **a²+1** deoarece **s--** decrementează variabila **s** după afișare!

10. **Răspuns corect:** d) 53078520

Indicații: În cazul de față funcția recursivă denumită **afis** se va opri în momentul în care **k** devine **1**. De menționat că, în cazul de față, **k** pornește de la valoarea **8** care este valoarea sumei indicilor lui **a[4][4]**, elementul din dreapta jos.

Valoarea variabilei **k** scade cu **2** la fiecare reapelare a funcției. Elementele se vor afișa din partea dreaptă spre cea stângă.

$$v = \begin{pmatrix} v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \\ v_{31} & v_{32} & v_{33} & v_{34} \\ v_{41} & v_{42} & v_{43} & v_{44} \end{pmatrix} \quad V = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 0 & 1 \\ 2 & 3 & 4 & 5 \end{pmatrix}$$

11. **Răspuns corect:** f) $\underbrace{f(t) \circ \dots \circ f(t)}_c$

Indicații: **abc(funcție(t), c-1)** apelează subprogramul **funcție()** de **c** ori. Această apelare se face cu rezultatul primit deja de la subprogramul **funcție()**, adică **f(f(x))**.

12. **Răspuns corect:** d) **cuei**

Indicații:

Pentru C++: A fost definit un tablou bidimensional de caractere, astfel:

	0	1	2	3	4	5	6	7	8	9	10
1	b	a	c	a	L	a	u	r	e	a	t
2	l	i	c	e	U						
3	e	x	a	m	E	n	e				
4	p	o	l	i	T	e	h	n	i	c	a

Prin urmare sunt afișate elementele:

i=1	a[1][2]='c'
i=2	a[2][4]='u'
i=3	a[3][6]='e'
i=4	a[4][8]='i'

Pentru Pascal: În comparație cu C++ unde șirul de caractere pornește de la 0, aici pornește de la 1. Prin urmare avem următoarele date:

	1	2	3	4	5	6	7	8	9	10	11
1	b	a	c	a	L	a	u	R	e	a	T
2	l	i	c	e	U						
3	e	x	a	m	E	n	e				
4	p	o	l	i	T	e	h	N	i	c	a

Următoarele elemente sunt afișate:

i=1	a[1][3]='c'
i=2	a[2][5]='u'
i=3	a[3][7]='e'
i=4	a[4][9]='i'

13. **Răspuns corect:** a) 17 5

Indicații: m=6, n=3 rezultă că după apelarea f1(m,n) vor fi următoarele valori: f1(6,3)=10, m=6, n=2. Apelarea f1(f1(m,n),m) se face, de fapt, pentru f1(10,6) care conduce la valorile f1(10,6)=17, m=5, n=2.

14. **Răspuns corect:** e) $b = \begin{pmatrix} 29 & 38 & 47 \\ 38 & 50 & 62 \\ 47 & 62 & 77 \end{pmatrix}$

Indicații:

i=1	j=1	k=1	b[1][1]=4
i=1	j=1	k=2	b[1][1]=13
i=1	j=1	k=3	b[1][1]=29
i=1	j=2	k=1	b[1][2]=6
i=1	j=2	k=2	b[1][2]=18
i=1	j=2	k=3	b[1][2]=38
i=1	j=3	k=1	b[1][3]=8
i=1	j=3	k=2	b[1][3]=23
i=1	j=3	k=3	b[1][3]=47

Identice se procedează și pentru i=2 și i=3.

15. **Răspuns corect:** e) 24

Indicații: n=4 rezultă că variabila b din funcție devine 12. Dacă b devine 12 rezultă că și n devine 12. Instrucțiunea a=2*a; atribuie lui a valoarea 2*n iar n este 12, adică 24.

Varianta 27

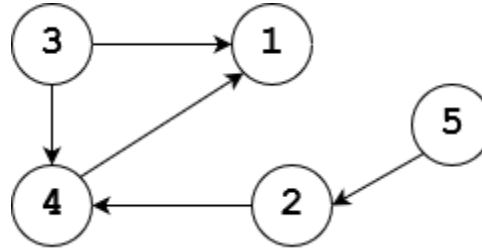
Indicații și răspunsuri

1. **Răspuns corect:** b) **-24**
Indicații: Se evaluează mai întâi rezultatul din paranteză, care este **-2**, apoi se calculează câtul împărțirii lui **16** la **-2** și în final acesta se înmulțește cu **3**.
2. **Răspuns corect:** b) **$n / 10 \% 10$** în varianta C/C++, respectiv **$n \text{ div } 10 \text{ mod } 10$** în varianta Pascal.
EXEMPLU: Pentru **$n=3185$** , **$n/10 = n \text{ div } 10 = 318$** , iar **$n/10\%10 = 318\%10 = 8$** , respectiv **$n \text{ div } 10 \text{ mod } 10 = 318 \text{ mod } 10 = 8$** .
3. **Răspuns corect:** f) La final **d1** și **d2** vor fi egale **doar** dacă **n** reține un număr pătrat perfect.
EXEMPLE: Pentru **$n=11$** vom avea **$d1=1$** și **$d2=11$** , deci **$d1 \neq d2$** , pentru **$n=125=5^3$** vom avea **$d1=5$** și **$d2=25$** , deci **$d1 \neq d2$** , pentru **$n=21$** vom avea **$d1=3$** și **$d2=7$** , , deci **$d1 \neq d2$** . Pentru orice pătrat perfect de forma **$n=r^2$** , vom avea **$d1=d2=r$** . De exemplu, pentru **$n=36$** vom avea **$d1=d2=6$** .
Indicații: Algoritmul reține la final în **d1** cel mai mare divizor al lui **n** al cărui pătrat e mai mic sau egal cu **n**, iar în **d2** divizorul „complementar”, care înmulțit cu **d1** dă rezultatul **n**. Doar în cazul pătratelor perfecte **$d1=d2$** .
4. **Răspuns corect:** a) **for (j = n - 2; j >= i; j--)** în varianta C/C++, respectiv **for j := n-2 downto i do** în varianta Pascal.
EXEMPLU: Dacă **$n=4$** și **$v=(3,2,1,4)$** , la primul pas, în care **$i=0$** vor fi comparate **1** și **4**, fără a fi nevoie de interschimbare, apoi **2** și **1**, care vor fi interschimbate și la final **3** și **1**, care vor fi de asemenea interschimbate. Tabloul va deveni astfel **(1,3,2,4)**. Pentru **$i=1$** vor fi comparate **2** și **4** fără a fi interschimbate, apoi **3** și **2**, care vor fi interschimbate. Tabloul va deveni **(1,2,3,4)**. Pentru **$i=2$** vor mai fi comparate, fără a fi interschimbate **3** și **4**.
Indicații: Algoritmul se aseamănă foarte mult cu sortarea prin metoda bulelor (Bubble Sort), constând de asemenea din parcurgeri succesive ale tabloului, în cadrul cărora sunt comparate și eventual interschimbate elemente aflate pe poziții consecutive. În cazul său însă parcurgerile se fac de la dreapta la stânga, în cadrul fiecăreia elementul minim fiind mutat în stânga subsecvenței prelucrate.
5. **Răspuns corect:** c) **$f(1, n) == n$** în varianta C/C++, respectiv **$f(1, n) = n$** în varianta Pascal.
Indicații: Apelul **$f(d, n)$** returnează cel mai mic divizor al lui **n** mai mare strict ca **d**. În cazul în care **n** este prim acesta va fi **n** și reciproc (în enunț se garantează că **n** este natural, mai mare strict ca **1**).
6. **Răspuns corect:** e) **36**
Indicații: Un arbore cu **10** vârfuri are cu siguranță **9** muchii, iar un graf complet cu **10** vârfuri are cu siguranță **$10 \cdot (10-1) / 2 = 45$** de muchii. Prin urmare, vor trebui adăugate **$45-9 = 36$** de muchii.
7. **Răspuns corect:** e) **65**
Indicații: Secvența de instrucțiuni construiește următoarea matrice:

25	24	23	22	21
20	19	18	17	16
15	14	13	12	11
10	9	8	7	6
5	4	3	2	1

8. **Răspuns corect:** c) (2, 1, 0, 2, 0)

Indicații: Dacă numerotăm vârfurile ca în figura alăturată, vom obține șirul gradelor interne de la punctul c)



9. **Răspuns corect:** c) Algoritmul generează în ordine lexicografică anagramele cuvântului citit care nu au vocale pe poziții alăturate.
Indicații: Ordinea alfabetică a literelor din cuvintele **pilo** și **poli** este **i, l, o, p**. Prin urmare varianta a) nu are cum să fie cea corectă (este înaintea lui **pilo** în ordine invers lexicografică). Variantele b) și d) nu sunt corecte, deoarece în ambele cazuri, penultimul cuvânt generat ar fi **poil**.
10. **Răspuns corect:** a) `strcpy(t, strchr(s, ' '))`; în varianta C/C++, respectiv `t:=copy(s, pos(' ', s), 5)`; în varianta Pascal.
Indicații: Variantele b), d), e) și f) ar produce erori de compilare, iar c) ar face ca în variabila **t** să fie reținut "Politehnica 20202020".
11. **Răspuns corect:** b) 31
Indicații: Ordinea apelurilor recursive va fi: $f(24, 34) = 2 + f(25, 34) = 2 + (1 + f(26, 33)) = 3 + f(26, 31) = 3 + f(26, 29) = 3 + (1 + f(27, 28)) = 4 + f(27, 26) = 4 + 27 = 31$
12. **Răspuns corect:** f) 3
Indicații: Putem elimina muchiile {1, 3}, {3, 4} și {3, 5}. Vom avea 3 cicluri elementare: (1, 5, 6, 1), (2, 5, 6, 2) și (1, 5, 2, 6, 1).
13. **Răspuns corect:** d) 0 1 1 1 3 5
Indicații: Primii trei vectori în ordine lexicografică sunt: (0, 1, 1, 1, 2, 5), (0, 1, 1, 1, 3, 5) și (0, 1, 1, 1, 4, 5). Vectorul (0, 1, 2, 3, 1, 1) nu este al doilea, (0, 1, 1, 1, 1, 3) corespunde unui arbore cu înălțimea 2, iar (0, 1, 1, 1, 2, 6) nu corespunde unui arbore (6 nu poate fi propriul său tată).
14. **Răspuns corect:** b) $m * m \geq x$
EXEMPLU: Dacă $x=30$ și apelurile subprogramului sunt următoarele (făcând observația că rezultatul dorit nu poate fi mai mare ca x): $rad(1, 30, 30) = rad(1, 15, 30) = rad(1, 8, 30) = rad(5, 8, 30) = rad(6, 8, 30) = rad(6, 7, 30) = rad(6, 6, 30) = 6$
Indicații: Subprogramul folosește o versiune modificată a căutării binare pentru a obține rezultatul dorit. Astfel dacă elementul din mijlocul intervalului în care se caută rezultatul are proprietatea dorită (pătratul său e mai mare sau egal cu x), căutarea continuă în prima jumătate, altfel în cea de-a doua.
15. **Răspuns corect:** d) `s -= v[j]`; în varianta C/C++, respectiv `s := s-v[j]`; în varianta Pascal.

EXEMPLU: Dacă $n=8$, $v=(3, 5, 4, 1, 2, 8, 19, 3)$ și $t=10$ atunci $lmax$ va avea la sfârșitul executării secvenței de instrucțiuni valoarea 3. Mai precis pentru $i=0$ se va găsi $j=0$, pentru $i=1$ se va găsi de asemenea $j=0$, deoarece suma primelor 2 elemente ale tabloului nu depășește $t=10$, pentru $i=2$ se va găsi $j=1$, deoarece suma primelor 3 elemente depășește t , dar $v[1]+v[2] \leq t$, pentru $i=3$ se va găsi $j=1$, pentru $i=4$ se va găsi $j=2$, pentru $i=5$ se va găsi $j=4$, pentru $i=6$ se va găsi $j=7$, pentru $i=7$ se va găsi $j=7$,

Indicații: Pentru fiecare i cuprins între 0 și $n-1$, algoritmul determină în timp liniar cel mai mare j cu proprietatea că subsecvența de la poziția i până la poziția j inclusiv respectă restricția de a avea suma mai mică sau egală cu t . Pentru verificarea acestei condiții este păstrată în permanență în variabila s suma elementelor din subsecvența curentă. Incrementarea lui j este echivalentă cu eliminarea lui $v[j]$ din subsecvență, prin urmare este necesar ca s să scadă cu $v[j]$.

Varianta 28

Indicații și răspunsuri

1. **Răspuns corect:** d) $(a+b+c+d) * 0.25$

Indicații: Varianta **a)** ar fi corectă doar dacă suma **a+b+c+d** ar fi scrisă între paranteze. Variantele **b)**, **c)**, **e)** și **f)** sunt incorecte din punct de vedere matematic, deși nu au erori de sintaxă.

2. **Răspuns corect:** c) Atât **S1**, cât și **S2**

Indicații: Ambele secvențe de instrucțiuni au ca efect obținerea în variabila **p** a primei cifre a numărului reținut inițial de **n**. În cadrul **S1** **p** reține mai întâi o copie a lui **n**, iar apoi, cât timp **p** are mai mult de o cifră se elimină ultima dintre acestea. În **S2** se reține în mod repetat în **p** ultima cifră a lui **n**, apoi aceasta se elimină prin păstrarea în **n** a câtului împărțirii întregi a lui **n** la **10**. În felul acesta ultima cifră eliminată este evident prima cifră a numărului reținut inițial de **n**, aceasta fiind valoarea finală a lui **p**. Singura diferență dintre **S1** și **S2** este că în cazul celei de-a doua secvențe de instrucțiuni valoarea inițială a lui **n** se va pierde, aceasta fiind înlocuită cu **0** la sfârșitul executării buclei.

3. **Răspuns corect:** f) $n \% d == 0$ în varianta C/C++, respectiv $n \bmod d = 0$ în varianta Pascal.

EXEMPLU: Pentru **n=300**, când **d=2**, în bucla **while** ($n \% d == 0$) a programului C/C++, respectiv **while** $n \bmod d = 0$ **do** a variantei Pascal **n** va deveni **75**, iar **p** și **nr** vor deveni **1**. Apoi, pentru **d=3**, în aceeași buclă **n** va deveni **25**, **p**, din nou **1**, iar **nr** va primi valoarea **2**. Pentru **d=4** nu se va intra în bucla **while** internă, deci **p** va rămâne **0**, iar **n** și **nr** nu se vor modifica. În fine, când **d=5**, **p** va deveni din nou **1**, **nr** va deveni **3**, iar **n** va ajunge **1**, ceea ce va face ca bucla **while** exterioară să se încheie.

Indicații: Algoritmul găsește divizorii primi ai lui **n** bazându-se pe faptul că, dacă atunci când e găsit un divizor **d**, **n** e împărțit la el de câte ori e posibil, atunci nu se va intra în bucla interioară decât pentru **divizorii primi** ai numărului reținut inițial de **n**.

4. **Răspuns corect:** e) $v[i+1] = x$ în varianta C/C++, respectiv $v[i+1] := x$ în varianta Pascal.

EXEMPLE: Dacă **n=5**, **v=(2,3,5,5,8)** și **x=4**, la primul pas, în care **i=4** sunt comparate **4** și **8**, iar **v[5]** devine **8**, apoi **i** devine **3** și, pentru că **v[3]=5** și e în continuare mai mare ca **x=4**, **v[4]** devine **5**. Același lucru se întâmplă pentru **i=2** și **v[3]** devine de asemenea **5**. Buclea se încheie cu **i=1**, pentru că **v[1]=3** și **3** nu e mai mare strict ca **x=4**. Evident, **4** va fi inserat după **3**, pe poziția **i+1 = 2**. Vectorul va deveni astfel **(2,3,4,5,5,8)**.

Dacă **n=4**, **v=(2,3,5,5)** și **x=1**, atunci în bucla **while** vor fi mutate cu o poziție spre dreapta toate elementele vectorului, iar la final **i** va avea valoarea **-1**. Pe poziția **i+1=0** va fi scris **x=1**. Vectorul va deveni astfel **(1,2,3,5,5,8)**.

Indicații: Algoritmul îl inserează pe **x** în **v** după ce mută spre dreapta cu o poziție toate componentele mai mari strict ca **x**. Buclea **while** se poate termina fie atunci când toate elementele lui **v** sunt strict mai mici ca **x** și sunt mutate (**i** devine **-1**), fie când se

ajunge la un i cu proprietatea că $v[i] \leq x$. În ambele cazuri este necesar ca x să fie inserat pe poziția $i+1$.

5. **Răspuns corect:** b) 60.

Indicații: Secvența de instrucțiuni construiește următoarea matrice:

5	4	3	2	1
6	5	4	3	2
7	6	5	4	3
8	7	6	5	4
9	8	7	6	5

6. **Răspuns corect:** b) f (225)

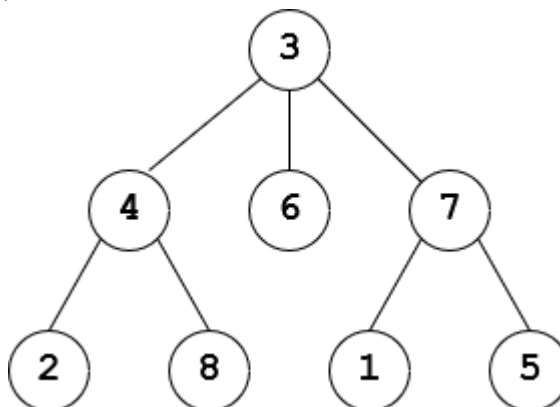
Indicații: În cadrul buclei **while** r devine cel mai mic număr al cărui pătrat este mai mare sau egal cu n (partea întreagă superioară a rădăcinii pătrate a lui n). Subprogramul returnează astfel 0 dacă și numai dacă n este pătrat perfect.

7. **Răspuns corect:** d) (1, 1, 2, 2)

Indicații: Variantele a) și f) sunt incorecte, pentru că un vârf nu poate avea gradul intern mai mare decât 3. Varianta b) are suma gradelor interne mai mare decât suma gradelor externe, iar varianta e) are suma gradelor interne mai mică decât suma gradelor externe. În cazul variantei c) contradicția provine din faptul că vârful 4 are gradul intern 0, deci nu poate exista un arc (1, 4), iar 1 are gradul extern 3. Graful cu arcele (1, 2), (1, 3), (1, 4), (2, 1), (3, 4), (4, 3), corespunde șirului gradelor interne (1, 1, 2, 2).

8. **Răspuns corect:** c) 5

Indicații: Arborele din figura alăturată corespunde vectorului de tați dat.

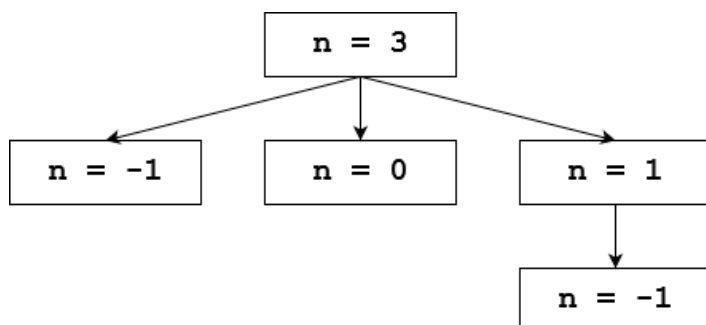


9. **Răspuns corect:** a) **strcpy(p, p+1)** în varianta C/C++, respectiv **delete(s, p, 1)** în varianta Pascal.

Indicații: Având în vedere că de fiecare dată caracterul c este căutat de la începutul șirului ($p = \text{strchr}(s, c)$), respectiv $p := \text{pos}(c, s)$) de caractere s , e necesar ca la fiecare iterație a buclei **while** să se șteargă ultima apariție găsită.

10. **Răspuns corect:** f) 012013

Indicații: Arborele apelurilor recursive este cel din figura de mai jos.



Apelul cu $n=3$ va afișa pe ecran **012**, apoi apelul cu $n=1$ va afișa **01** și în final apelul cu $n=3$ va mai afișa **3**. Celelalte **3** apeluri nu vor afișa nimic, întrucât condiția $n>0$ nu va fi verificată în cazul lor.

11. **Răspuns corect:** b) Generarea tuturor permutărilor mulțimii $\{1, 2, \dots, n\}$.

Indicații: În cazul ambelor probleme elementul curent din vectorul soluție, $sol[p]$ trebuie să respecte condiția $sol[p] \neq sol[i]$ pentru orice $i < p$.

12. **Răspuns corect:** d) G nu poate fi eulerian

Indicații: Un graf cu **10** vârfuri și mai puțin de **9** muchii nu poate fi conex, deci afirmația de la punctul a) este adevărată. Un graf care nu este conex nu poate fi hamiltonian, deci afirmația de la punctul c) este adevărată. Graful G poate avea două cicluri elementare cu câte **4** vârfuri și **4** muchii și **2** vârfuri izolate (**4** componente conexe în total), deci afirmația de la punctul b) este adevărată. În fine, un graf eulerian poate avea vârfuri izolate, deci în cazul lui G putem avea un ciclu cu **8** vârfuri care să conțină toate cele **8** muchii și alte două vârfuri izolate. Prin urmare, afirmația de la punctul d) este singura care **nu** e adevărată.

13. **Răspuns corect:** e) **5**

Indicații: O soluție posibilă este un arbore cu vectorul de tați:

$(0, 1, 1, 1, 2, 2, 2, 5, 5, 5, 8, 11)$ cu lanțul $(1, 2, 5, 8, 11, 12)$ de lungime **5**, care unește rădăcina **1** cu frunza **12**.

14. **Răspuns corect:** a)

```

if (v[n-2] > v[n-1])
{
    aux = v[n-1];
    v[n-1] = v[n-2];
    v[n-2] = aux;
    mysort(n - 1, v);
}
  
```

pentru limbajul C/C++, respectiv

```

if v[n-2] > v[n-1] then
    begin
        aux := v[n-1];
        v[n-1] := v[n-2];
        v[n-2] := aux;
        mysort(n-1, v)
    end
  
```

pentru limbajul Pascal

Indicații: În urma apelului recursiv $mysort(n-1, v)$, primele $n-1$ componente ale lui v vor fi ordonate. Dacă $v[n-2] \leq v[n-1]$, atunci nu mai sunt necesare alte

prelucrări. În caz contrar ultimele două componente ale subsecvenței ($v[0]$, $v[1]$, ..., $v[n-2]$, $v[n-1]$) care trebuie ordonată vor fi interschimbate, iar procesul va fi reluat pentru primele $n-1$ elemente printr-un nou apel recursiv cu aceleași argumente. Cum $v[n-1]$ este cu siguranță elementul maxim după interschimbare, la revenirea din apelul recursiv, subsecvența va fi ordonată.

15. **Răspuns corect:** c) $v[i] > s[m-1]$

EXEMPLU: Dacă $n=8$, $v=(3,3,1,8,2,1,5,4)$, corespunzător lui $x=33182154$ și $k=4$ atunci aplicând algoritmul vom obține vectorul $s=(8,2,5,4)$, corespunzător lui $Y=8254$. Acesta e cel mai mare număr natural care poate fi obținut din x prin eliminarea a exact 4 cifre, fără a schimba ordinea în care cifrele apăreau în x .

Indicații: Algoritmul folosește vectorul s , organizat după principiul „ultimul sosit – primul servit”. Componentele acestuia sunt acele cifre care pot face parte din Y . În momentul luării în considerare a unei noi cifre ($v[i]$) aceasta este introdusă cu siguranță în s (nu se știe încă nimic despre cifrele care urmează și deci acestea ar putea fi mai mici). Apariția cifrei curente ($v[i]$) poate avea ca efect eliminarea din s a altor cifre, mai mici, care nu trebuie să se afle înaintea lui $v[i]$ în Y . Aceste eliminări se fac în bucla **while** prin decrementarea lui m (numărul de elemente din vectorul s).

Varianta 29

Indicații și răspunsuri

1. **Răspuns corect:** c) $(n-1)/2$

EXEMPLU: Pentru $n=7$ și tabloul unidimensional (4 2 3 1 6 8 5) se obține tabloul (5 8 6 1 3 2 4).

Indicații: Elementul de pe poziția din mijloc nu este necesar să fie mutat. Se interschimbă 4 cu 5, 2 cu 8 și 3 cu 6, 3 interschimbări, adică se fac $(n-1)/2$ interschimbări.

2. **Răspuns corect:** b) (1 15 102 18 91)

Indicații: Se mută al treilea element spre stânga cu 2 poziții ș.a.m.d. Primele 2 elemente se mută în aceeași ordine la sfârșit.

3. **EXEMPLU:**

	x		
			x
x			
		x	

Răspuns corect: c) 3 1 4 2

Indicații: Două dame nu se atacă între ele dacă nu se află pe aceeași coloană, pe aceeași diagonală sau pe același rând. Astfel așezarea lor poate fi: dama 1 pe rândul 3, dama 2 pe rândul 1, dama 3 pe rândul 4, dama 4 pe rândul 2.

4. **Răspuns corect:** b) 10

EXEMPLU: Pentru tabloul unidimensional (10 24 9 11 33 7 15) se fac următoarele interschimbări:

Pasul 1: (10 9 11 24 7 15 33): 4 interschimbări

Pasul 2: (9 10 11 7 15 24 33): 3 interschimbări

Pasul 3: (9 10 7 11 15 24 33): 1 interschimbare

Pasul 4: (9 7 10 11 15 24 33): 1 interschimbare

Pasul 5: (7 9 10 11 15 24 33): 1 interschimbare

Indicații: O interschimbare se face dacă sunt îndeplinite simultan condițiile $v[i] > v[j]$ și $i < j$.

5. **Răspuns corect:** f) 26789 34567 34568 34569 34578

Indicații:

a) 45678 45679 45689 45789 46789 56789 – lipsește numărul subliniat

b) 34789 35678 35679 35689 36789 45678 – lipsește numărul subliniat

c) 34578 34569 34568 34567 26789 – nu sunt în ordine crescătoare

d) 13458 13459 13467 13469 13478 13479 – lipsește numărul subliniat

e) 13458 13459 13467 13468 13469 – numărul subliniat are mai mult de 2 cifre alăturate de aceeași paritate

6. **Răspuns corect:** c) 2349

Indicații: Subprogramul parcurge recursiv cele două numere **a** și **b** și returnează un număr format cu cifra mai mare de pe aceeași poziție din cele două numere.

7. **Răspuns corect:** d) 3

Indicații: Numărul minim de comparații se obține folosind algoritmul de căutare binară. Comparația 1: elementul din mijloc: 73; Comparația 2: elementul din mijloc: 95; Comparația 3: elementul din mijloc: 82;

8. **Răspuns corect:** e) nedefinită
Indicații: Este o variabilă locală neinițializată.
9. **Răspuns corect:** b) 190
Indicații: În total sunt 400 de elemente, 20 sunt pe diagonala secundară.
10. **Răspuns corect:** d) 1023
EXEMPLU: Fie cele 3 tije **a**, **b** și **c**. Se mută discurile de pe tija **a** pe tija **b**, utilizând ca tijă intermediară tija **c**. Pentru **k=3** discuri se fac 7 mutări: **a→b**, **a→c**, **b→c**, **a→b**, **c→a**, **c→b**, **a→b**.
Indicații: Pentru **k** discuri este necesar un număr de $2^k - 1$ mutări.
11. **Răspuns corect:** c) 8
Indicații: Graful nu conține bucle. Fiecare vârf este adiacent cu toate celelalte vârfuri. Dacă graful are **n** vârfuri există **n (n-1)** arce. **56=n (n-1)**, **n=8**.
12. **Răspuns corect:** f) Doi dintre algoritmi nu diferă ca eficiență din punctul de vedere al timpului de executare.
Indicații:
a) Algoritmii A_1 și A_2 rezolvă problema pentru orice date de intrare cu valorile din interval.
b) Algoritmul A_2 este cel mai eficient din punctul de vedere al timpului de executare. Are complexitatea $O(n)$ pentru parcurgerea șirului și determinarea sumei.
c) Algoritmul A_4 are complexitate dată de o sortare rapidă, $O(n \log n)$.
d) Algoritmul A_4 rezolvă problema.
e) Algoritmii rezolvă problema pentru orice date de intrare cu valorile din interval.
f) A_3 și A_4 au aceeași eficiență, complexitate dată de o sortare rapidă, $O(n \log n)$.
Observație: Există algoritmi care rezolvă această problemă într-un timp mai scurt (folosind operații pe biți).
13. **Răspuns corect:** f) E_1 , E_2 și E_3
14. **Răspuns corect:** c) 5040
EXEMPLU: Anagramele (nu neapărat în această ordine) sunt: **aaccerrt**, **aaccetrtr**, **aaccetrtr**, ..., **rtreccaa**, **trreccaa**. În total 5040.
Indicații: Litera **a** apare de două ori, litera **c** apare de două ori, litera **e** apare o dată, litera **r** apare de două ori, litera **t** apare o dată; permutările sunt cu repetiție.
Numărul permutărilor este: $8! / (2! \cdot 2! \cdot 1! \cdot 2! \cdot 1!)$
15. **Răspuns corect:** b) doar formula 1
Indicații:
Formula 1 poate fi obținută din ecuația

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$$
Claculăm determinantul: $(-1)^n = F_{n+1} * F_{n-1} - F_n^2$.
Știm că $M^m * M^n = M^{m+n}$, pentru orice matrice M pătratică.
Așadar $(-1)^m = F_{m+1} * F_{m-1} - F_m^2$.

$$M^m = \begin{pmatrix} F_{m+1} & F_m \\ F_m & F_{m-1} \end{pmatrix}$$

$$M^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$$

$$M^{m+n} = \begin{pmatrix} F_{m+n+1} & F_{m+n} \\ F_{m+n} & F_{m+n-1} \end{pmatrix}$$

$$\begin{aligned} M^m \cdot M^n &= \begin{pmatrix} F_{m+1} & F_m \\ F_m & F_{m-1} \end{pmatrix} \cdot \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \\ &= \begin{pmatrix} F_{m+1} \cdot F_{n+1} + F_m \cdot F_n & F_{m+1} \cdot F_n + F_m \cdot F_{n-1} \\ F_m \cdot F_{n+1} + F_{m-1} \cdot F_n & F_m \cdot F_n + F_{m-1} \cdot F_{n-1} \end{pmatrix} = M^{m+n} = \\ &= \begin{pmatrix} F_{m+n+1} & F_{m+n} \\ F_{m+n} & F_{m+n-1} \end{pmatrix} \end{aligned}$$

Identificăm:

$$F_m \cdot F_{n+1} + F_{m-1} \cdot F_n = F_{m+n} \quad (a)$$

$$F_m \cdot F_n + F_{m-1} \cdot F_{n-1} = F_{m+n-1} \quad (b)$$

Punem $m = n$

$$F_n \cdot F_{n+1} + F_{n-1} \cdot F_n = F_{2n} \quad (a)$$

$$F_n^2 + F_{n-1}^2 = F_{2n-1} \quad (b)$$

Din relația (a) rezultă

$$F_{2n} = F_n \cdot (F_{n+1} + F_{n-1}) = F_n \cdot (F_{n-1} + F_n + F_{n-1}) = F_n \cdot (2 \cdot F_{n-1} + F_n).$$

Așadar:

Dacă n este par, înjumătățindu-l, rezultă:

$$F(n) = F\left(\frac{n}{2}\right) \cdot \left[2 \cdot F\left(\frac{n}{2} - 1\right) + F\left(\frac{n}{2}\right)\right].$$

Dacă n este impar, din (b), rezultă:

$$F(n) = \left[F\left(\frac{n+1}{2}\right)\right]^2 + \left[F\left(\frac{n-1}{2}\right)\right]^2.$$

Termeni, conform formulelor, sunt:

1, 1, 2, 3, 5, 8, ...

Referitor la formulele **2** și **3**, corect este: $F(n) = \frac{1}{\sqrt{5}} \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n$

Varianta 30

Indicații și răspunsuri

1. **Răspuns corect:** c) $n/2$
EXEMPLU: Pentru $n=8$ și tabloul unidimensional (4 2 3 1 9 6 8 5) se obține tabloul (5 8 6 9 1 3 2 4).
Indicații: Se interschimbă 4 cu 5, 2 cu 8, 3 cu 6 și 1 cu 9, 4 interschimbări, adică se fac $n/2$ interschimbări.
2. **Răspuns corect:** d) 0 elemente
Indicații: Nu este nevoie de spațiu de memorie suplimentar.
3. **Răspuns corect:** f) 0
Indicații: Nu există nicio soluție pentru 3 dame.
4. **Răspuns corect:** b) 3 5
Indicații: Matricea de adiacență are numărul de linii egal cu numărul de coloane.
5. **Răspuns corect:** e) $dc(x, y) = dc(y, x \bmod y)$
EXEMPLU: pentru $x=6$ și $y=8$
 a) $dc(6, 8) \neq dc(48, 8)$
 b) $dc(6, 8) \neq dc(6, 6)$
 c) $dc(6, 8) = dc(8, 48) = dc(48, 384) = \dots$
 d) $dc(6, 8) \neq dc(6, 6)$
 f) $dc(6, 8) \neq dc(0, 0)$
Indicații: Formula pentru a calcula cel mai mare divizor comun folosind algoritmul lui Euclid este $dc(x, y) = dc(y, x \bmod y)$. Se folosește recursiv până se obține restul 0.
6. **Răspuns corect:** b) 2 subtablouri, nu întotdeauna cu același număr de elemente. În partea stângă se vor găsi elementele mai mici decât pivotul, în partea dreaptă se vor găsi elementele mai mari decât pivotul, elementele egale cu pivotul pot rămâne în oricare parte a pivotului. Rezultă că cele două părți nu au întotdeauna același număr de elemente.
7. **Răspuns corect:** e) stivă
EXEMPLU: Se creează o stivă, operația de adăugare numită *push()*, memorând șirul de la primul caracter până la ultimul. Astfel fiecare caracter al șirului va fi, pe rând, în vârful stivei. Extragerea se face din vârful stivei, eliminând, pe rând, primul element, folosind operația numită *pop()*.
Indicații: Stiva corespunde principiului *LIFO* (Last In First Out).
8. **Răspuns corect:** d) $f(n/2)$;(respectiv $f(n \text{ div } 2)$)
Indicații:
 a) $f(n-2)$ subprogramul nu se încheie pentru valori impare ale lui n .
 b) $f(n-1)$ subprogramul nu se încheie pentru valori negative ale lui n .
 c) $f(n\%2)$ (respectiv $f(n \bmod 2)$) subprogramul nu se încheie pentru valori impare ale lui n .
 e) $f(n+2)$ subprogramul nu se încheie pentru valori ale lui n diferite de -2 și 0 .
 f) $f(n*2)$ subprogramul nu se încheie pentru valori nenule ale lui n .
9. **Răspuns corect:** e) 4
Indicații: Un apel $f(1)$ are rezultatul 2, celălalt apel $f(1)$ are rezultatul 2
10. **Răspuns corect:** d) 28

Indicații: Folosim principiul includerii-excluderii. Considerăm mulțimea $M = \{101, 102, \dots, 200\}$. Luăm alte trei submulțimi **A**, **B**, **C** care au ca elemente cele aflate în **M** divizibile cu 2, 3, respectiv 5. Determinăm elementele comune mulțimilor **A** și **B**, **A** și **C**, **B** și **C**, apoi comune mulțimilor **A**, **B** și **C**.

EXEMPLU: Pentru valorile date avem: $\text{card}M=100$. $\text{card}A=50$. $\text{card}B=33$. $\text{card}C=20$.
 $\text{card}(AB)=17$. $\text{card}(AC)=10$. $\text{card}(BC)=7$. $\text{card}(ABC)=3$.
 $\text{card}M - \text{card}A - \text{card}B - \text{card}C + \text{card}(AB) + \text{card}(AC) + \text{card}(BC) - \text{card}(ABC) = 100 - 50 - 33 - 20 + 17 + 10 + 7 - 3 = 28$.

11. **Răspuns corect:** a) **xx**

Indicații: Dacă programul generează permutări de elemente care se repetă atunci cele două caractere **x** nu sunt diferite; se formează o singură permutare.

12. **Răspuns corect:** d) **3**

Indicații: sunt adevărate enunțurile 2, 3 și 5.

Enunțul 1: se generează **28** de numere cu prima cifră **2**.

Enunțul 2: **12457, 12459, 12479, 12679, 14679, 34679**.

Enunțul 3: **13679** sau **24568**.

Enunțul 4, un număr corect este **12789**.

Enunțul 5: cifra **1** apare de 50 de ori pe prima poziție, cifra **9** apare de 50 de ori pe ultima poziție.

13. **Răspuns corect:** d) $(n-1)!/2$

Indicații: În graful neorientat complet orice permutare a celor **n** noduri este un ciclu hamiltonian. Un același ciclu, ca permutare circulară, se parcurge în $2 \cdot n$ moduri. Se pornește de la un nod în sensul acelor de ceasornic, dar și în sens trigonometric.

Exemplu: pentru **n=4**,

Cicluri identice:

(1 2 3 4 1), (1 4 3 2 1), (2 3 4 1 2), (2 1 4 3 2), (3 4 1 2 3), (3 2 1 4 3), (4 1 2 3 4), (4 3 2 1 4);

(1 3 4 2 1), (1 2 4 3 1), (2 4 3 1 2), (2 1 3 4 2), (3 4 2 1 3), (3 1 2 4 3), (4 2 1 3 4), (4 3 1 2 4);

(1 4 2 3 1), (1 3 2 4 1), (2 4 1 3 2), (2 3 1 4 2), (3 1 4 2 3), (3 2 4 1 3), (4 1 3 2 4), (4 2 3 1 4).

În total avem 3 cicluri distincte hamiltoniene.

În general există $n! / (2 \cdot n) = (n-1)! / 2$ cicluri hamiltoniene distincte.

14. **Răspuns corect:** b) **5**

Indicații: Numărul de grafuri orientate complete cu **n** noduri este $3^{C_n^2}$, adică $3^{n(n-1)/2}$.

$3^{n(n-1)/2} = 59049$, $3^{n(n-1)/2} = 3^{10}$, **n=5**.

15. **Răspuns corect:** c) doar relațiile E_1 și E_2

Indicații:

E_1 : Se poate observa că fiecare al treilea termen din șirul lui Fibonacci este par.

E_2 : $F_n = F_{n-1} + F_{n-2} = (F_{n-2} + F_{n-3}) + (F_{n-3} + F_{n-4}) = F_{n-2} + 2 \cdot F_{n-3} + F_{n-4} = (F_{n-3} + F_{n-4}) + 2 \cdot F_{n-3} + F_{n-4} = 3 \cdot F_{n-3} + F_{n-4} + (F_{n-5} + F_{n-6}) = 3 \cdot F_{n-3} + F_{n-3} + F_{n-6} = 4 \cdot F_{n-3} + F_{n-6}$.

Așadar, cum fiecare al treilea termen din șirul lui Fibonacci este par, atunci F_{n-3} și F_{n-6} sunt ambii pari. Notăm F_n cu $F_p(n)$, rezultă că F_{n-3} este termenul par precedent, adică $F_p(n-1)$, iar F_{n-6} este $F_p(n-2)$.
 Înseamnă că $F_p(n) = 4 \cdot F_p(n-1) + F_p(n-2)$, $n \geq 2$, $F_p(0) = 0$ și $F_p(1) = 2$.

Varianta 31

Indicații și răspunsuri

1. **Răspuns corect:** d) 2024
Indicații: Expresia are valoare maximă dacă $n=8080$ deoarece 8080 este cel mai mare număr natural de patru cifre multiplu de 2020.
 $2020 - n \% 2020 + n / 2020 = 2020 - 0 + 4 = 2024$
2. **Răspuns corect:** a) 234
Indicații: Nu căutăm decât cazurile când $y=2$. Pentru $x=0$ și $y=2$ se afișează 2; pentru $x=1$ și $y=2$ se afișează 3; pentru $x=2$ și $y=2$ se afișează 4.
3. **Răspuns corect:** f) 42
Indicații: Se observă că toate elementele situate pe linia 2 sunt egale cu 1 și toate elementele situate pe linia 4 sunt egale cu 3. Suma elementelor situate pe linia 3 este 6 și suma elementelor situate pe linia 5 este 12. În rest, elementele din tabloul bidimensional au valoarea 0.
 $1 * 6 + 3 * 6 + 2 * 3 + 4 * 3 = 42$
4. **Răspuns corect:** a) AUTONATICA
Indicații: Se elimină a doua literă O din BUTONOMATICA, apoi se înlocuiește litera B cu litera A, apoi se elimină litera M.
5. **Răspuns corect:** b) Limbajul C++/C $(I.b \leq J.a) \ || \ (J.b \leq I.a)$
Limbajul Pascal $(I.b \leq J.a) \ or \ (J.b \leq I.a)$
Indicații: Sunt două cazuri când I și J nu se intersectează: fie când extremitatea dreaptă a lui I este mai mică sau egală cu extremitatea stângă a lui J, fie când extremitatea dreaptă a lui J este mai mică sau egală cu extremitatea stângă a lui I.
6. **Răspuns corect:** b) 12
Indicații: Numerele căutate sunt pare deci au pe ultimele două poziții 00, 02, 10, 12, 20, 22. Cifra sutelor poate să fie 1 sau 2 pentru că numerele sunt naturale și au exact 3 cifre. Deci sunt $2 * 6 = 12$ numere.
7. **Răspuns corect:** c) 3080
Indicații: Pentru rapiditate se poate utiliza formula $n(n+1)(n+2)/3$, unde $n=20$.
Deci $20 * 21 * 22 / 3 = 3080$.
8. **Răspuns corect:** b) (16, 9, 7, 5, 4, 3, 2, 1, 0)
Indicații: Primele 5 numere naturale pătrate perfecte: 0, 1, 4, 9, 16. Sortate descrescător ajung: 16, 9, 4, 1, 0.
Primele 4 numere naturale prime: 2, 3, 5, 7. Sortate descrescător ajung: 7, 5, 3, 2.
După interclasare, C are 9 elemente și următorul conținut: (16, 9, 7, 5, 4, 3, 2, 1, 0).
9. **Răspuns corect:** a) 19 20 20
Indicații: Numai c își schimbă valoarea.
10. **Răspuns corect:** c) 514
Indicații: Toate nodurile de la 1 și până la 512 au câte doi descendenți direcți(fii). Nodul 513 are un singur fiu: 1026; toate nodurile de la 514 și până la 1026 sunt frunze.
 $1026 - 512 = 514$
11. **Răspuns corect:** c) 9
Indicații: O soluție mai rapidă se bazează pe calculul invers: se scad cele șase subgrafuri care au mulțimea muchiilor nevidă din 15 (numărul total de subgrafuri ale grafului neorientat cu 4 noduri, $2^4 - 1 = 15$). Subgrafurile cu două muchii au mulțimea nodurilor

$\{1, 2, 3, 4\}$, respectiv $\{1, 2, 3\}$ (două cazuri). Subgrafurile cu o muchie au mulțimea nodurilor $\{1, 2, 4\}$, $\{1, 2\}$, $\{2, 3, 4\}$, respectiv $\{2, 3\}$ (patru cazuri).

$$2^4 - 1 - 2 - 4 = 9$$

12. **Răspuns corect:** b) $O(\log n)$, algoritm logaritmic

Indicații: Se utilizează un algoritm logaritmic bazat pe calculul sumei

$$[n/5] + [n/(5*5)] + [n/(5*5*5)] + \dots$$

13. **Răspuns corect:** e) 1

Indicații: Arcul pe care este suficient să îl adăugăm este $(2, 4)$.

14. **Răspuns corect:** c) 4^9

Indicații: Numărul total de grafuri neorientate cu 8 noduri este $2^{8*(8-1)/2}$

Numărăm ce nu variază deasupra diagonalei principale din matricea de adiacență asociată grafului deoarece graful este neorientat.

Numărăm perechile de noduri adiacente. Sunt 4 cazuri:

$$[2, 8], [3, 8], [5, 8], [7, 8]$$

Numărăm perechile de noduri neadiacente. Sunt 6 cazuri:

$$[1, 3], [1, 5], [1, 7], [3, 5], [3, 7], [5, 7].$$

$$\text{Deci numărul căutat este } 2^{8*(8-1)/2} - 4 - 6 = 2^{28-10} = 2^{18} = 4^9$$

15. **Răspuns corect:** e) 8

Indicații: Tabloul unidimensional memorează răsturnatele primelor 33 de numere naturale pătrate perfecte. Deoarece căutăm câte numere din tablou se termină cu cifra 1, problema se rezumă la identificarea acelor pătrate perfecte care încep cu cifra 1. Nu ne interesează numerele pătrate perfecte mai mari decât 200 și mai mici decât 999 pentru că nu încep cu cifra 1.

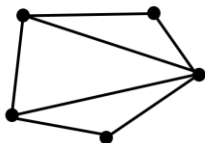
Pătrate perfecte căutate: 1, 16, 100, 121, 144, 169, 196, 1024, deci sunt 8 numere.

Varianta 32

1. **Răspuns corect: c)**
2. **Răspuns corect: a)**
Indicații: Variabila **i** reține, în ordine descrescătoare, multiplii comuni ai variabilelor **a** și **b**. La final variabila **i** reține cel mai mic multiplu comun al acestora.
3. **Răspuns corect: a)**
Indicații: Pentru **x=9** se afișează: **18 27 36 45 54 63 72 81 90**.
4. **Răspuns corect: d)**
5. **Răspuns corect: c)**
Indicații: se formează trei componente conexe, două cu câte trei noduri și una cu patru noduri.
6. **Răspuns corect: b)**
Indicații: Algoritmul lui Euclid reprezintă o metodă eficientă de calculare a celui mai mare divizor comun a două numere întregi.
7. **Răspuns corect: d)**
Indicații: Graful este conex și toate gradele sunt pare.
8. **Răspuns corect: c)**
9. **Răspuns corect: b)**
Indicații: Submulțimile generate sunt: **{2} {1,2} {2,5} {2,9} {1,2,5} {1,2,9} {2,5,9} {1,2,5,9}**.
10. **Răspuns corect: c)**
11. **Răspuns corect: c)**
Indicații: Pe nivelul **1** este un nod care are doi fii; pe nivelul **2** sunt **2** noduri care au, fiecare câte **3** fii; pe nivelul **3** sunt **6** noduri care au, fiecare câte **4** fii.
Deci, numărul de frunze este **24**.
12. **Răspuns corect: e)**
Indicații: În antetul subprogramului **f**, **y** este parametru formal transmis prin referință.
13. **Răspuns corect: f)**
Indicații: Se calculează cmmdc al numerelor **x** și **y** și apoi cmmdc pentru numărul rezultat și **z**.
14. **Răspuns corect: a)**
15. **Răspuns corect: b)**
Indicații: Ciclurile care sunt aceleași cu excepția punctului de plecare nu sunt luate în calcul separat.

Varianta 33

1. Răspuns corect: a)
2. Răspuns corect: f)
3. Răspuns corect: d)
4. Răspuns corect: d)
5. Răspuns corect: c)



Indicații:

6. Răspuns corect: b)

0	3	6	9	2	5	8	1	4	7
2	5	8	1	4	7	0	3	6	9
4	7	0	3	6	9	2	5	8	1
6	9	2	5	8	1	4	7	0	3
8	1	4	7	0	3	6	9	2	5
0	3	6	9	2	5	8	1	4	7
2	5	8	1	4	7	0	3	6	9
4	7	0	3	6	9	2	5	8	1
6	9	2	5	8	1	4	7	0	3
8	1	4	7	0	3	6	9	2	5

Indicații:

7. Răspuns corect: b)

Indicații: $\frac{100(100-1)}{2} = 4950$.

8. Răspuns corect: e)

Indicații: $\frac{b+c}{2} + \frac{c+d}{2} = \frac{b+d}{2} + c$.

9. Răspuns corect: c)

Indicații: Graful este conex și toate gradele sunt pare; nu există lanț hamiltonian.

10. Răspuns corect: c)

Indicații: Soluțiile sunt: **3+4+5** și **3+9**.

11. Răspuns corect: a)

Indicații: Arborele are **n-1** muchii și suma gradelor unui graf este dublul numărului de muchii.

12. Răspuns corect: b)

13. Răspuns corect: c)

Indicații: La fiecare pas se adaugă **3k** și se scot **k+2**, adică rămân în coadă **2k+2** elemente. După **9** pași numărul de elemente este **2 (1+2+...+9) + 2*9=90-18=72**.

14. Răspuns corect: a)

15. Răspuns corect: f)

Indicații: Arborele are **n-1** muchii. Numărul de elemente nule din matricea de adiacență este: **n²-2 (n-1) = n²-2n+2**.

Varianta 34

1. Răspuns corect: a)

2. Răspuns corect: c)

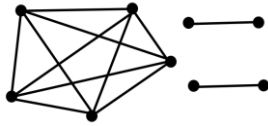
3. Răspuns corect: c)

Indicații: În timpul rulării, variabilei **b** are următoarele valori:

0 1 2 4 7 8 9 15 22 23 24.

4. Răspuns corect: a)

5. Răspuns corect: c)



Indicații:

6. Răspuns corect: a)

7. Răspuns corect: c)

8. Răspuns corect: d)

9. Răspuns corect: d)

10. Răspuns corect: c)

11. Răspuns corect: a)

Indicații: În antetul subprogramului **f**, **b** este parametru formal transmis prin referință.

12. Răspuns corect: b)

Indicații: Graful complet cu **n** noduri are $\frac{n(n-1)}{2}$ muchii.

13. Răspuns corect: c)

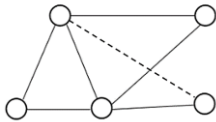
14. Răspuns corect: c)

Indicații: $S_1 = S_2 = \frac{n(n-1)}{2}$.

15. Răspuns corect: c)

Varianta 35

1. Răspuns corect: b)
2. Răspuns corect: d)
3. Răspuns corect: b)
Indicații: Pentru fiecare $i \in \{1, 2, 3, \dots, 10\}$, se realizează 10 execuții ale instrucțiunii care afișează valoarea variabilei k . Numărul final de execuții este: $10 \cdot 9 = 90$.
4. Răspuns corect: c)
5. Răspuns corect: c)
6. Răspuns corect: d)
7. Răspuns corect: e)
Indicații: sunt necesare $11(11-1)/2 = 55$ comparații și $11(11-1)/2 = 55$ interschimbări.
8. Răspuns corect: b)
9. Răspuns corect: f)
Indicații: (1 2 3 4) (1 4 3 2) (3 2 1 4) (3 4 1 2)
10. Răspuns corect: c)
11. Răspuns corect: e)
Indicații: În antetul subprogramului f , y este parametru formal transmis prin referință
12. Răspuns corect: b)



Indicații:

13. Răspuns corect: a)
14. Răspuns corect: d)
15. Răspuns corect: b)

Varianta 36

1. Răspuns corect: f) 288 1

Indicații:

Operatori aritmetici binari multiplicativi	Limbajul Pascal	Limbajul C/C++
Înmulțirea	*	*
Câtul împărțirii întregi	div	/
Restul împărțirii întregi	mod	%

Cei trei operatori au aceeași prioritate și se evaluează de la stânga la dreapta.
 $x=288$, $y=1$.

2. Răspuns corect: c) ($x=y$ și $y \neq z$) sau ($x \neq y$ și $y=z$)

Indicații: Cele trei variabile trebuie să fie inițializate. Expresia este **1/True** dacă cele trei variabile sunt egale sau diferite două câte două. Expresia este **0/False** dacă oricare două variabile sunt egale și oricare două diferite.

3. Răspuns corect: e) 100

Indicații: $x=y=100$. Deoarece condiția $x>y$ este falsă se execută $y \leftarrow 10*x-8*y$. Valoarea lui y se modifică, $y=200$. Diferența absolută $|x-y|=|y-x|=100$.

4. Răspuns corect: c) $a \geq 1$

Indicații: Limbajul C++/C: Secvențele date sunt echivalente atunci când **<condiția>** din instrucțiunea **while** este la fel cu **<condiția>** din instrucțiunea **do... while**.

Limbajul Pascal: Secvențele date sunt echivalente atunci când **<condiția>** din instrucțiunea **while... do** devine **<negație condiție>** în instrucțiunea **repeat... until**.

5. Răspuns corect: d) 96

Indicații: $f(5)=2*f(4)=2*(2*f(3))=2*2*(2*f(2))=2*2*2*(2*f(1))=2*2*2*2*(2*f(0))=2*2*2*2*(2*3)=96$

6. Răspuns corect: d) strcat | concat

Indicații: Concatenarea a două șiruri se poate realiza în **Limbajul C++/C** cu subprogramul predefinit **strcat**, iar în **Limbajul Pascal** cu funcția predefinită **concat**.

7. Răspuns corect: a) 3

Indicații: Există 3 lanțuri distincte de lungime 3 de la nodul 1 la nodul 4:
 $L1=[1,2,3,4]$, $L2=[1,2,5,4]$ și $L3=[1,5,2,4]$.

8. Răspuns corect: b) 1

Indicații: Primul nod este rădăcina arborelui. Fiecare nod are un singur descendent. Ultimul nod este frunză.

9. Răspuns corect: d) 207

Indicații: Numerele generate sunt: 108, 126, 153, 162, 18, 207 etc.

10. Răspuns corect: f) 2043231

Indicații: Elementele tabloului sunt: $a[0]=1$, $a[1]=2$, $a[2]=3$...
 $a[2020]=2021$.

Suma elementelor este $s = \frac{n(n+1)}{2}$. Pentru $n=2021$ suma elementelor este $\frac{2021*2022}{2} = 2043231$.

11. Răspuns corect: e) 5

5	6	10	20	1	tabloul inițial
---	---	----	----	---	-----------------

Indicații: La fiecare parcurgere se compară elementele învecinate și se realizează interschimbul doar între elementele care nu respectă relația de ordine. Configurația este finală atunci când nu se mai realizează niciun interschimb.

5	6	10	1	20	prima parcurgere
5	6	1	10	20	a 2-a parcurgere
5	1	6	10	20	a 3-a parcurgere
1	5	6	10	20	a 4-a parcurgere
1	5	6	10	20	a 5-a parcurgere

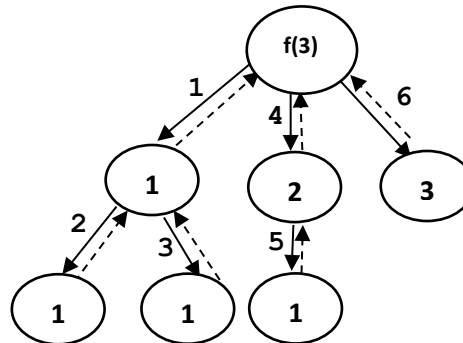
12. Răspuns corect: b) 29

Indicații: Numere excepționale:

110, 111, 112, 113, 114, 115, 116, 117, 118, 19, 210, 310, 410, 510, 610, 710, 810, 910, 221, 331, 242, 441, 551, 661, 771, 283, 881, 392, 991.

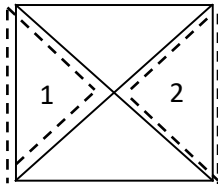
13. Răspuns corect: f) 3

Indicații: În schema apelurilor recursive, pentru $n=3$, valorile afișate sunt încercuite: 1 1 1 2 1 3. Numerele asociate săgeților indică ordinea de executare a apelurilor recursive și, implicit, ordinea de afișare a valorilor.



14. Răspuns corect: c)

Indicații: Secvența interschimbă elementele triunghiurilor **unu** și **doi** inclusiv elementele de pe diagonale, celelalte elemente păstrându-și poziția inițială.



Matricea inițială

1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5

Matricea finală

5 2 3 4 1
5 4 3 2 1
5 4 3 2 1
5 4 3 2 1
5 2 3 4 1

15. Răspuns corect: d) 252

Indicații: Pentru n număr natural format din 3 cifre, subprogramul va returna un număr natural format din aceleași 3 cifre. Dacă n conține cel puțin o cifră de 9, atunci numărul returnat va avea cifra sutelor egală cu 9.

În intervalul [100, 199] există 19 numere naturale care au cel puțin o cifră de 9. În

intervalul [100, 899] există $8 \cdot 19 = 152$ numere naturale care au cel puțin o cifră de 9.

În intervalul [900, 999] există 100 de numere care au cel puțin o cifră de 9.

Așadar, pentru $n \in [100, 999]$ subprogramul poate returna 252 numere naturale cu cifra sutelor 9.

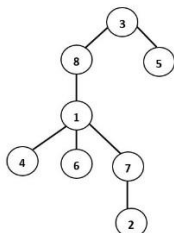
Varianta 37

1. Răspuns corect: b)

Indicații: Se face diferența la nivel de cod ASCII ($97 - 99 = -2$).

2. Răspuns corect: c)

Indicații: Reprezentarea grafică a respectivului arbore este:



3. Răspuns corect: f)

Indicații: Matricea de adiacență este simetrică față de diagonala principală prin urmare numărul total de cifre 1 este $n(n-1) = n^2 - n$. Cum pe diagonala principală avem un număr de n cifre de 0, iar numărul total de elemente din matrice este n^2 , matricea desemnează încă de la bun început un graf norientat complet.

4. Răspuns corect: c)

Indicații: Pentru a obține un număr maxim de noduri izolate, vom asigura totalul de 10589 de muchii cu un număr minim de noduri (147). Cum restul nodurilor sunt izolate, gradul maxim pe care îl poate avea un nod din cele 147 este egal cu 146.

5. Răspuns corect: f)

Indicații: Pentru a ajunge la cuvântul din mijloc, se elimină primul cuvânt din șirul de caractere, iar apoi în cadrul structurii repetitive următoarele patru cuvinte. Ultimul cuvânt copiat în variabila **c** desemnează cuvântul căutat.

6. Răspuns corect: b)

Indicații: Variabila **k** va desemna numărul seriei de termeni în care se găsește cel de pe poziția **n**, iar **s** va reține poziția ultimului termen din acea serie.

7. Răspuns corect: c)

Indicații: Ex: pentru $n=5$, se va construi în memorie un tablou simetric față de ambele diagonale de forma:

```

2 3 4 5 6
3 4 5 6 5
4 5 6 5 4
5 6 5 4 3
6 5 4 3 2
  
```

8. Răspuns corect: c)

Indicații: Se rețin în ordine descrescătoare multiplii comuni ai celor două variabile. La final variabila **d** va păstra valoarea celui mai mic multiplu comun al acestora.

9. Răspuns corect: d)

Indicații: Variabila **c** reține puterea lui 5 din factorialul lui **a**, dar cum există în același timp și un număr de elemente pare mai mare decât **c**, valoarea sa va desemna și numărul de 0 obținut din înmulțiri de forma $2 \cdot 5$.

10. Răspuns corect: c)

Indicații: Pentru a ajunge la o anumită literă din șirul de caractere **strada**, va trebui mai întâi accesat un element din tablou (**v[5]**), iar apoi câmpul **adresa**.

11. Răspuns corect: c)

Indicații: La prima deschidere a fișierului se vor citi toate valorile existente în acesta, iar la cea de-a doua se vor citi doar primele **n-1** valori care includ și valoarea citită initial în **n**.

12. Răspuns corect: a)

Indicații: Se caută valoarea raportului dintre suma cifrelor (obținută prin apelul **f(n)**) și numărul de cifre calculat în variabila **c**.

13. Răspuns corect: e)

Indicații: Se lipesc primele două caractere din șirul **p** la ceea ce a rămas în **s** după eliminarea de caractere.

14. Răspuns corect: b)

Indicații: Secvența parcurge în spirală în sensul acelor de ceasornic elementele tabloului respectiv.

15. Răspuns corect: e)

Indicații: Pentru a obține media dorită, variabila **ev** nu este validată dacă apelul funcției **medie** nu se regăsește în intervalul de valori [**x-0.5**; **x+0.5**).

Varianta 38

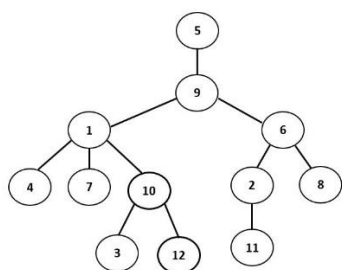
1. Răspuns corect: f)

Indicații: Termenii șirului lui Fibonacci: $1, 1, 2, 3, 5, 8, 13, 21, 34, 55, \dots$

Primii cinci termeni impari diferiți duc la suma: $1+3+5+13+21= 43$

2. Răspuns corect: b)

Indicații: Reprezentarea grafică a respectivului arbore este:



3. Răspuns corect: d)

Indicații: Numărul minim de încercări este obținut la depistarea parolei încă de la prima testare, iar numărul maxim este dat de formula număr total caractere

$(52+10= 62)$ număr caractere parolă

4. Răspuns corect: b)

Indicații: Muchia $[1, 6]$ va deveni muchia $[3, 6]$.

5. Răspuns corect: f)

Indicații: Calculul valorii variabilei d se va opri în momentul în care variabila i va ajunge la valoarea 0.

6. Răspuns corect: f)

Indicații: Elementele de pe poziții pare nu vor primi valori din fișier și prin urmare elementul $v[8]$ are valoarea 0 obținută din declararea vectorului ca parametru global.

7. Răspuns corect: d)

Indicații: Se vor afișa elementele tabloului pe 4 linii și 3 coloane respectând formula de calcul $i+j$.

8. Răspuns corect: e)

Indicații: Variabila k va trece prin toți termenii de la 0 la 10, iar variabila p va dezvolta pentru fiecare termen k înmulțirile cu $0, 1, \dots, 10$.

9. Răspuns corect: a)

Indicații: Secvența va șterge pe rând fiecare apariție a subșirului **test** în ordinea apariției acestora în șirul inițial

10. Răspuns corect: c)

Indicații: Secvența folosește formula de calcul matematic al produsului dintre două matrice.

11. Răspuns corect: e)

Indicații: Variabila c va parcurge toate caracterele aflate între literele mici m și r , dar la afișare se vor trece cele ce ocupa 5 poziții în urmă, respectiv **hijklm**.

12. Răspuns corect: a)

Indicații: Secvența duce în prima parte la răsturnarea caracterelor din cadrul șirului de caractere, dar acestea vor fi repositionate în formatul inițial în a doua jumătate a instrucțiunii repetitive.

13. Răspuns corect: d)

Indicații: Se citesc pe rând datele corespunzătoare celor trei elevi, iar în paralel în **s** se calculează suma tuturor notelor din fișierul de intrare. Dacă din variabila **s** se scade valoarea 71, se va obține suma notelor lui Sebby, prin urmare expresia afișată reprezintă media acestuia.

14. Răspuns corect: a)

Indicații: Dacă diferența dintre componenta de pe poziția curentă și cea anterioară nu respectă rația dintre primele două componente, variabila **ev** va primi valoarea 0 ce indică o valoare invalidă.

15. Răspuns corect: c)

Indicații: Se parcurg în paralel cele două diagonale și se interschimbă fiecare element de pe diagonala principală ($a_{i, i}$) cu fiecare element de pe cea secundară ($a_{i, n-i+1}$).

Varianta 39

Indicații și răspunsuri

1. **Răspuns corect:** c) a și -1
Indicații: Din numărul total de valori din șir, 100, elimină numărul cifrelor din șir
2. **Răspuns corect:** e) VBPRE
3. **Răspuns corect:** e) patru
Indicații: Primele patru elemente din tablou primesc valoarea 9
4. **Răspuns corect:** c) 2
Indicații: după prima parcurgere 51 ajunge pe poziția finală, după cea de-a doua parcurgere 40 ajunge pe poziția finală
5. **Răspuns corect:**
Limbajul C++/C b) $(i < j) \ \&\& \ (i + j < n + 1)$
Limbajul Pascal b) $(i < j) \ \text{AND} \ (i + j < n + 1)$
Indicații: Condiția stabilește o intersecție pe cele două zone: zona aflată deasupra diagonalei principale $(i < j)$ și zona aflată deasupra diagonalei secundare $(i + j < n + 1)$
6. **Răspuns corect:** e) 12
Indicații: Graful neorientat cu 8 noduri și 28 de muchii este un graf complet. Pentru un număr minim de muchii eliminate se aleg 2 componente conexe astfel: o componentă conexă cu 2 noduri și o componentă conexă cu 6 noduri, se vor elimina $6 + 6 = 12$ muchii.
7. **Răspuns corect:**
Limbajul C++/C a) $x * y > y * z \ \&\& \ x * z > y * z$
Limbajul Pascal a) $(x * y > y * z) \ \text{AND} \ (x * z > y * z)$
8. **Răspuns corect:**
Limbajul C++/C b) $(x > 1000) \ \&\& \ ((x * x * x) \% 1000 == 0)$
Limbajul Pascal b) $(x > 1000) \ \text{AND} \ ((x * x * x) \text{ MOD } 1000 = 0)$
Indicații: $x = 36 * 35 = 1260$
9. **Răspuns corect:** f) $\lceil \log_2 n \rceil + 1$
10. **Răspuns corect:** d) 45
11. **Răspuns corect:** c) este conex și suma elementelor de pe fiecare coloană a matricei de adiacență este număr par
12. **Răspuns corect:** a) 8 7 20 12
13. **Răspuns corect:** e) 12600
Indicații: $C_{10}^1 * C_9^2 * C_7^3 * C_4^4 = 10 * 36 * 35 * 1 = 12600$ șiruri distincte
14. **Răspuns corect:** b) 8
Indicații: Pentru fiecare nod ales drept nod rădăcină, există un singur vector de tați
15. **Răspuns corect:** b) verifică dacă numărul x este divizibil cu b-1
Indicații: Se aplică criteriul de divizibilitate: un număr natural scris în bază b se divide cu b-1 dacă și numai dacă suma cifrelor sale este un multiplu de b-1.

Varianta 40

Indicații și răspunsuri

1. **Răspuns corect:** d) 63
Indicații: $f(63)=f(62)+63=\dots=f(4)+5+6+\dots+63=8+(5+6+\dots+63)=2014$
2. **Răspuns corect:** e) 50
3. **Răspuns corect:** c) studentarterou
Indicații: La primul șir se concatenează cel de-al doilea șir, mai puțin primul caracter, apoi se concatenează cel de-al treilea șir, mai puțin primele două caractere.
4. **Răspuns corect:** d) oricare ar fi x, y, z , p egal cu q
Indicații: cele două expresii sunt echivalente
5. **Răspuns corect:** c) 130
6. **Răspuns corect:** a) $[-2^{n-1}, 2^{n-1}-1]$
Indicații: primul bit, din reprezentare, este cel de semn (0- pentru numere întregi pozitive și 1- pentru numere întregi negative), ceilalți $n-1$ biți sunt folosiți pentru reprezentarea valorii absolute a numărului.
7. **Răspuns corect:** b) 16
Indicații: se intră o singură dată în instrucțiunea **while**, variabila **p** nu se modifică
8. **Răspuns corect:** b) 7, 16, 10
Indicații: se înjumătățește secvența curentă în care se face căutarea
9. **Răspuns corect:** e) 777
Indicații: Numărul valorilor de 1 din tabloul bidimensional, pe linii, este $1 + 2 + 4 + 8 + 16 + 32 + 64 + 128 = 255$. Tabloul are 8 linii și 129 coloane. De unde, numărul de valori 0 este: $(129-1) + (129-2) + \dots + (129-128) = 777$
10. **Răspuns corect:** a) 377
Indicații: se pot folosi termenii din șirul lui Fibonacci
11. **Răspuns corect:** c) 11
Indicații: Graful neorientat are 20 de muchii care formează o componentă conexă folosind 7 noduri. Rămân 10 noduri izolate.
12. **Răspuns corect:** b) 13
Indicații: 2 și toate numerele impare cuprinse între 3 și $\lceil \sqrt{681} \rceil$
13. **Răspuns corect:** d) 101
Indicații: Numărul total de permutări cu 5 elemente este $5!=120$. După permutarea 51423 se mai generează încă 19 termeni.
14. **Răspuns corect:** b) A, B
Indicații: $c.m.m.m.c(m,n)=m*n/c.m.m.d.c(m,n)$
15. **Răspuns corect:** f) 4^{13}
Indicații: Numărul grafurilor neorientate cu 8 noduri este 2^{28} . Numărul grafurilor neorientate cu 8 noduri, în care nodurile 2 și 3 sunt neadiacente, este $2^{28}/2=2^{27}$, s.a.m.d.

Variantă 41

Indicații și răspunsuri

1. **Răspuns corect:** a) exact $n(n-1)/2 - m$
Indicații: Graful G_1 va conține muchiile grafului complementar al lui G (dacă în G există o muchie în G_1 nu va exista). Din graful complet cu n noduri și $n(n-1)/2$ muchii se scad muchiile grafului G , m
2. **Răspuns corect:** f) $x \leq d$
Indicații: Funcția descompune în factori primi un număr. Când numărul x devine egal cu divizorul atunci numărul x este un factor prim la puterea 1
3. **Răspuns corect:** d) 4
Indicații: Se parcurge vectorul pentru căutarea valorii x , repetiția oprindu-se la prima apariție a valorii x în vector sau după ce toate elementele au fost parcurse dacă x nu apare în vector.
4. **Răspuns corect:** d) 128
Indicații: Există șirul de apeluri: $F(7), F(6) \dots F(0)$ care duce la valorile returnate de la stânga la dreapta 1, $2*1, 2*2, 2*4, 2*8, 2*16, 2*32, 2*64=128$
5. **Răspuns corect:** c) CDEFGEFG
Indicații: Se memorează șirul de caractere începând de la poziția 4 (C) | 5 (Pascal) apoi șirul începând de la poziția 2 (C) | 3 (Pascal) și apoi se concatenează aceste două șiruri.
6. **Răspuns corect:** d) B,C
Indicații: La evaluare se ține cont de prioritatea operatorilor
7. **Răspuns corect:** b) BEC BED CAB
Indicații: Se pleacă de la variantele propuse și aplicând metoda backtracking se generează următoarele soluții.
8. **Răspuns corect:** a) de 5 ori
9. **Răspuns corect:** f) 0
Indicații: Fiecare nod n are ca fii nodurile $2n$ și $2n+1$. Fiind număr impar nu rămâne nici un nod cu un fiu.
10. **Răspuns corect:** e) $n(n-1)/2$
Indicații: Inițial vectorul este ordonat crescător, deci se face numărul maxim de interschimbări.
11. **Răspuns corect:** c) $n=5, U=\{[1,3], [1,4], [3,4], [2,4], [4,5], [2,5]\}$
Indicații: Se desenează fiecare graf. Ca să fie eulerian trebuie ca să existe un ciclu care să conțină toate muchiile grafului o singură dată iar ca să nu fie hamiltonian trebuie să nu existe niciun ciclu care să conțină toate nodurile grafului o singură dată.
12. **Răspuns corect:**
Limbajul C++/C b)

```
if(x>y && y>z)
    p=x*y*z;
```


Limbajul Pascal b)

```
if (x>y) AND (y>z)
    then p:=x*y*z;
```


Indicații: Din proprietatea de tranzitivitate se observă că expresia logică $z>x$ are întotdeauna valoare 1 | true.

13. **Răspuns corect:** b) cuprins între 7 și 12
Indicații: Numărul de înjumătățiri este $\log 1000$
14. **Răspuns corect:** a) A
Indicații: Se folosește algoritmul care modifică numărul prin adunarea ultimei cifre la câtul împărțirii la 10 până când se obține un număr care are o singură cifră.
15. **Răspuns corect:** c) 94
Indicații: $1+3+3x2+3x2^2+3x2^3+3x2^4=94$

Varianta 42

Indicații și răspunsuri

1. **Răspuns corect:** d) 6
Indicații: Subprogramul se apelează pe prima jumătate (de la n la $(n+m)/2$) și pe a doua ($(n+m)/2 + 1$ până la m) adunând restul fiecărui element la imparitatea cu 2.
2. **Răspuns corect:** a) Numai S1
Indicații: S1 funcționează cât timp există litera în șir și cât timp caracterele sunt diferite de litere mari. Asemeni cel de-al doilea. Când se opresc afișează caracterul la care s-au oprit.
3. **Răspuns corect:** d) $1+2=3$
Indicații: Se trece peste primele două litere, când se ajunge la primul 1 se schimbă următorul cu caracterul +. Se trece la primul caracter de 2, următorul după acesta fiind schimbat în caracterul =, urmând ca mai apoi să rămână la final doar caracterul 3.
4. **Răspuns corect:** d) trei
Indicații: Se parcurg elementele până la întâlnirea lui 0, pentru $i > 0$.
5. **Răspuns corect:** f) Cea mai lungă secvență de valori de parități diferite
Indicații: Se parcurge vectorul numărând elementele consecutive de parități diferite. Dacă se ajunge la un element de aceeași paritate cu cel de dinainte se resetează lungimea secvenței și se analizează lungimea maximă.
6. **Răspuns corect:** f) 62
7. **Răspuns corect:** e) C_{n-p+1}^2
8. **Răspuns corect:** c) $j*j + \text{suma}(j*j-1)$
9. **Răspuns corect:** b) 1
Indicații: Se repetă o singură dată pentru că se evaluează $v[0]$, care este 0, deci repetiția se oprește, și apoi se incrementează i .
10. **Răspuns corect:** d) 2^5
Indicații: Pe nivelul 0, se găsește un nod (rădăcina), pe nivelul 1, 2 noduri, pe nivelul 2, 4 noduri, pe nivelul 3, 8 noduri, pe nivelul 4, 16 noduri iar nivelul 5 32 noduri.
11. **Răspuns corect:** d) 1023; 1032; 105; 1203;
Indicații: Se pornește de la o soluție propusă și se generează cu algoritmul backtracking celelalte soluții.
12. **Răspuns corect:** a) metoda căutării binare
Indicații: Sunt 2 metode de căutare: secvențială și binară. Eficientă este metoda căutării binare pentru care se efectuează $\log(n)$ operații.
13. **Răspuns corect:** d) permutărilor
Indicații: Se folosește metoda backtracking. Coloanele din matrice vor reprezenta elementele care se generează.
14. **Răspuns corect:** f) 20
Indicații: Vor exista doar muchii în care extremitatea inițială este mai mică decât extremitatea finală. Deci, dacă există drumul de la i la j , în care nodurile sunt în ordine crescătoare, nu va exista drumul de la j la i . Deci, fiecare nod formează o componentă tare conexă.

15. **Răspuns corect:** c) 13

Indicații: Se numără numărul de înjumătățiri care se fac pentru a ajunge la 1.

Varianta 43

Indicații și răspunsuri

1. **Răspuns corect:** b) 1

Indicații: Se execută operațiile matematice în ordinea priorității operatorilor.

2. **Răspuns corect:** e) 3.15

Indicații: Variabilelor i și j , fiind de tip întreg, li se atribuie $[x]$ și, respectiv, $[y]$.

3. **Răspuns corect:** d) 0

Indicații: Se evaluează $j \neq 0$ (F) și apoi operatorul de incrementare $=>i=1$. Pentru că $i \neq 0$ se decrementează $=>i=0$ și apoi se evaluează suma.

4. **Răspuns corect:** e) 51970

Indicații: Se introduc în x cifrele impare din n , în aceeași ordine, doar că se începe de la cifra zecilor.

5. **Răspuns corect:** f) 8

Indicații: Se calculează ultima cifră nenulă a numărului $n!$.

6. **Răspuns corect:** d) 28 56 4 13 6 18 26 90 25

Indicații: Se șterg elementele nule din tabloul unidimensional.

7. **Răspuns corect:** b) Informatica-poli

Indicații: Se determină adresa de memorie a caracterului '-', împărțindu-se astfel șirul inițial în două șiruri de caractere. Apoi se concatenează al doilea șir cu primul, după ce prima literă a fiecărui șir se transformă.

8. **Răspuns corect:** b) b

Indicații: Se declară un tablou unidimensional cu 2 elemente de tip structură, fiecare element conținând un pointer către un șir de caractere. Câmpului din a doua structură i se atribuie șirul definit de la poziția 1.

9. **Răspuns corect:** c) A și D

Indicații: Pentru a afla numărul de drumuri de lungime k dintre două noduri i și j într-un graf orientat, se calculează matricea $X=A^k$, unde A este matricea de adiacență. $X_{i,j}$

reprezintă numărul de drumuri. Deci $A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$ și $A^3 = \begin{pmatrix} 2 & 2 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 2 & 1 \\ 1 & 1 & 1 & 3 \end{pmatrix}$ Se

observă că între $A^3[1,3]=A^3[3][1]=0$.

10. **Răspuns corect:** d) 4

Indicații: Șirul apelurilor este: $f(0) \xrightarrow{0<7} f(2) \xrightarrow{2<7} f(4) \xrightarrow{4<7} f(6) \xrightarrow{6<7} f(8) \xrightarrow{8>7} 8-3$ raspuns care se returnează la apelul anterior, adunându-se de fiecare dată 1. $f(0)=7$. Apoi se calculează $f(7)=2$; $f(2)=6$ și $f(6)=4$.

11. **Răspuns corect:** c) 2 4 1 5 3
 5 1 4 2 3
 1 2 3 4 5
 5 4 3 2 1
 4 1 5 3 2

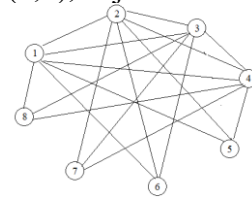
Indicații: Se fac 3 permutări circulare ale ultimelor 3 linii astfel încât matricea la final revine la configurația inițială.

12. **Răspuns corect:** c) 7196
 7198

Indicații: Dacă se pornește de la 7196 se continuă cu 7197 (incorectă, se repetă cifra 7), 7 1 9 8, se revine la a doua cifră și se alege 2, dar fiind prim nu e permis, la fel 3 și se ajunge la 4, apoi se continuă cu 0 și cu 1, adică 7401.

13. **Răspuns corect:** c) II și IV

Indicații: În secvența IV, există un nod cu gradul 8, ceea ce nu este posibil într-un graf cu 8 noduri. În secvența II, gradul 6 al celor 4 vârfuri înseamnă 18 muchii (nodul 1 incident cu 6 muchii, nodul 2 incident cu 5 muchii (a șasea fiind (1,2), deja numărată)) etc. Adică



$6+5+4+3=18$ muchii. Un exemplu posibil de graf ar fi:
 dată: $6+6+6+6+3+3+2+2=34 \Rightarrow 17$ (muchii contradicție).

Din secvența

14. **Răspuns corect:** b) $O(n)$ pentru f_1 și $O(2^n)$ pentru f_2

Indicații: Am putea scrie pentru fiecare funcție, complexitatea timp, recurent, astfel:

$T_1(n) = T_1(n-1) + C$ care este $O(n)$

$T_2(n) = 2 \cdot T_2(n-1) + C$ care este $O(2^n)$

15. **Răspuns corect:** a) $O(\log_2 k)$

Indicații: Se calculează n^k în timp logaritmic după următoarea metodă:

$$n^k = \begin{cases} n \cdot (n^2)^{\frac{k-1}{2}}, & \text{daca } k \text{ impar} \\ (n^2)^{\frac{k}{2}}, & \text{daca } k \text{ par} \end{cases}$$

Varianta 44

Indicații și răspunsuri

1. **Răspuns corect:** c) 10

Indicații: Se fac operațiile matematice în ordinea priorității operatorilor.

2. **Răspuns corect:** e) 3.5

Indicații: Se folosește operatorul cast, astfel încât rezultatul împărțirii i/j să fie real.

3. **Răspuns corect:** f) 5

Indicații: Se evaluează $j \neq 0$ (A) și apoi operatorul de decrementare $\Rightarrow j=2$. Pentru că $!j=0$ atunci se evaluează $i \neq 0$ (A) și se incrementează $j \Rightarrow j=3$.

4. **Răspuns corect:** d) 3

Indicații: Se concatenează la sfârșitul șirului s șirul ABCDE și se obține ABCDEABCDE, apoi se șterg primele 3 caractere. Deci șirul s este DEABCDE.

5. **Răspuns corect:** b) 1

Indicații: Apelul funcției implică decrementarea valorii parametrului și returnarea acestui rezultat.

6. **Răspuns corect:** a) $p^*=2 \mid p:=p*2$

Indicații: Fie $n=d_1^{e_1} \cdot d_2^{e_2} \cdot \dots \cdot d_r^{e_r}$, descompus în factori primi. Știm că numărul de divizori ai lui n este $(e_1 + 1) \cdot (e_2 + 1) \cdot \dots \cdot (e_r + 1)$. Folosindu-se acest rezultat, se determină factorii primi până la $\sqrt{n}$. Dacă $n \neq 1$ atunci n este un număr prim, deci la produsul anterior mai trebuie înmulțit 2.

7. **Răspuns corect:** e) $x[k/2]*(x[k/2]-1)/2 \mid x[k \text{ div } 2]*(x[k \text{ div } 2]-1) \text{ div } 2$

Indicații: Se folosește vectorul de frecvență x astfel încât, $x[i]$ să exprime numărul de elemente din v care au restul i la împărțirea cu k. Se observă că algoritmul nu adaugă numărul de perechi obținute între elemente cu același rest $[k:2]$, în cazul în care k este par. Acest număr reprezintă numărul de combinații ale valorilor de rest $[k:2]$.

8. **Răspuns corect:** c) $a[n-j-1][n-i-1]=2; \mid a[n-j-1,n-i-1]:=2;$

Indicații: Se parcurg elementele din cadranul 1 și apoi prin simetrie față de diagonala secundară se vor completa și elementele din cadranul II.

9. **Răspuns corect:** c) -1

Indicații: Șirul apelurilor este: $f(16) \xRightarrow{16>8} f(13) \xRightarrow{13>8} f(10) \xRightarrow{10>8} f(7) \xRightarrow{7<8} 7-5=2$. Se calculează $f(2)=-3$, apoi se revine cu $-3+4$, se calculează $f(1)=-4$ ș.a.m.d.

10. **Răspuns corect:** c) 8

Indicații: Se observă că între nodurile 1, 2, 6 există drum între oricare două noduri, deci ele formează o componentă tare conexă. Între celelalte 7 noduri nu există drum de la i la j și de la j la i, deci fiecare nod formează separat, câte o componentă tare conexă.

11. **Răspuns corect:** e) apnmdc

Indicații: Plecând de la soluția apnmdc se generează următoarele încercări: ~~apnmdd~~, ~~apnmddm~~, ~~apnmddn~~, ~~apnmddp~~... și se revine la primul caracter și se generează ebadid

12. **Răspuns corect:** b) se elimină o muchie și se adaugă două

Indicații: Se observă că graful are două componente conexe. Fiecare componentă conexă are 501 noduri, fiind subgrafuri complete. Deși gradul fiecărui nod este par, graful nu este eulerian pentru că nu este conex. Dacă, de exemplu, se elimină muchia [2,4] din subgraful cu noduri pare, atunci gradele celor două noduri devin impare, deci trebuie refăcută paritatea, așa că putem adăuga, de exemplu, muchia [2,1], dar și [4,1]. În felul acesta, graful devine conex și se păstrează paritatea gradelor nodurilor.

13. **Răspuns corect:** d) 7

Indicații: Pentru ca înălțimea arborelui să fie minimă, numărul de fii ai fiecărui nod trebuie să fie maxim, adică 4. Deci pe nivelul 0 se găsește rădăcina, pe nivelul 1 vor fi 4^1 noduri, pe nivelul 2, $4^2=16$ noduri, pe nivelul 3, $4^3=64$ noduri, apoi $4^4=256$, $4^5=1024$, pe nivelul 6, restul nodurilor. Deci, în total, 7 nivele.

14. **Răspuns corect:** d) $O(n \log n)$

Indicații: Pentru prima repetiție timpul este $O(n)$, dar pentru a doua, timpul este logaritm.

15. **Răspuns corect:** b) n

Indicații: Expresia se poate scrie astfel:

$$a_0 + a_1 * x + a_2 * x^2 + a_3 * x^3 + \dots + a_n * x^n = a_0 + x * (a_1 + x * (a_2 + \dots + x * (a_{n-1} + x * a_n) \dots))$$

Varianta 45

Indicații și răspunsuri

1. **Răspuns corect:** d) 15

Indicații: Întâi se evaluează operatorul de decrementare, deci valoarea variabilei **i** devine 3, se face produsul **i * j** și apoi se evaluează operatorul de incrementare.

2. **Răspuns corect:** c) -4

Indicații: Întâi se face produsul, deci **k=-9**, apoi la **k** se adună **j** => **k=-12** și se determină câtul împărțirii lui **k** la **i**.

3. **Răspuns corect:** b) -1

Indicații: Se evaluează **j≠0** și apoi operatorul de decrementare => **i=1**. Pentru că **i≠0** se mărește **j** cu 1 => **j=-1** și apoi se evaluează produsul.

4. **Răspuns corect:** d) 2 1

Indicații: Se returnează valoarea inițială a parametrului și apoi se incrementează parametrul, pentru că operatorul de incrementare este în formă postfixată. Deci se returnează 1 iar parametrul va avea valoarea 2.

5. **Răspuns corect:** e) 4

Indicații: Este declarat un vector cu 2 elemente de tip structură, fiecare element conținând un alt vector cu două numere întregi. Se atribuie valoare doar elementelor **S[0].a[1]** și **S[1].a[0]**.

6. **Răspuns corect:** f) 30

Indicații: Algoritmul determină toate numerele din intervalul [1, 10000] care sunt formate doar din cifrele 4 și/sau 6. Exemple 4, 6, 44, 46, 64, 66 etc.

7. **Răspuns corect:** c) II

Indicații: Se parcurg elementele din zona I, dar se folosesc elementele simetrice cu acestea față de diagonala secundară, deci elemente din zona II

8. **Răspuns corect:** b) Automatica-UPB

Indicații: Se determină adresa de memorie | poziția caracterului '-', împărțindu-se astfel șirul inițial în două șiruri de caractere. Apoi se concatenează al doilea șir cu primul, după ce prima literă a celui de al doilea șir se transformă în literă mare.

9. **Răspuns corect:** f) **a[i]>a[j]**

Indicații: Se folosește algoritmul de ordonare prin numărare. Fiecare element **b[i]** memorează numărul de elemente din tabloul unidimensional **a**, mai mari decât **a[i]**.

10. **Răspuns corect:** a) -6

Indicații: Șirul apelurilor este: $f(19,7) \xRightarrow{19>7} f(16,8) \xRightarrow{16>8} f(13,9) \xRightarrow{13>9} f(10,10) \xRightarrow{10=10} f(11,10) \xRightarrow{11>10} f(8,11) \xRightarrow{8<11} 3*8-2*11$ raspuns care se returnează la apelul anterior șamd, scăzându-se 2, unde e cazul.

11. **Răspuns corect:** c) 4316 3618 3418

Indicații: O abordare ar putea să plece de la o variantă dată. Folosind metoda backtracking, ținând cont de condițiile impuse de problemă, se pleacă de la primul număr propus și se generează următoarele trei soluții.

12. **Răspuns corect:** a) $2^{i-1} - 1$

Indicații: Pentru ca pe niveluri, să avem număr maxim de noduri, trebuie ca toate nodurile de pe nivelurile anterioare să aibă câte 2 fii. Deci, pe nivelul 1, avem 2^0 noduri, pe nivelul 2 sunt 2^1 noduri, pe nivelul 3 sunt 2^2 noduri ș.a.m.d. Deci, pe nivelul i vor fi 2^{i-1} noduri. Cum numărul total de noduri este par, iar pe nivelul 1 este un nod, atunci pe nivelul i vor fi $2^{i-1}-1$ noduri.

13. **Răspuns corect:** c) 25

Indicații: Partițiile mulțimii nodurilor pot avea:

prima mulțime – 1 nod; a doua - 9 noduri => nr de muchii $1*9$

prima mulțime – 2 noduri; a doua - 8 noduri => nr de muchii $2*8$

....

prima mulțime – 5 noduri; a doua - 5 noduri => nr de muchii $5*5$ (maxim)

14. **Răspuns corect:** b) $O(n)$

Indicații: Complexitatea algoritmului pare a fi $O(n^2)$. Pentru că a doua repetiție nu resetează valoarea lui j la 0, atunci vor fi valori ale lui i pentru care a doua repetiție nu se mai execută. Deci, vom avea maxim 2 treceri prin vector, deci $O(n)$.

15. **Răspuns corect:** e) 30

Indicații: Pentru a afla numărul de drumuri de lungime k dintre două noduri i și j într-un graf orientat, se calculează matricea $X=A^k$, unde A este matricea de adiacență. $X_{i,j}$ reprezintă numărul de drumuri dintre nodurile i și j de lungime k . Pentru a calcula numărul total de drumuri din digraf, atunci se adună toate valorile matricei X . Deci

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \dots, A^3 = \begin{pmatrix} 3 & 2 & 2 & 1 \\ 3 & 2 & 3 & 2 \\ 2 & 1 & 2 & 2 \\ 2 & 0 & 1 & 2 \end{pmatrix}. \text{ Se adună toate elementele și se obține 30.}$$

Varianta 46

Indicații și răspunsuri

1. **Răspuns corect:** c) 6

Indicații: Se fac operațiile în ordinea priorității operatorilor

2. **Răspuns corect:** e) 32

Indicații: Sunt generate toate numerele care conțin doar cifrele 5 și/sau 7.

3. **Răspuns corect:** d) 48 52 26 11 41 65

Indicații: Se rearanjează elementele vectorului, astfel încât cele pare să fie la începutul șirului, iar cele impare la sfârșit.

4. **Răspuns corect:** d) `informaticatest`

Indicații: Funcția `strtok` (C) separă șirul inițial în două șiruri. Se concatenează al doilea șir cu primul și se modifică șirul inițial. Nu se mai adaugă spațiu între cele două șiruri.

5. **Răspuns corect:** f) $r > t$

Indicații: Pentru intersecția a n intervale se determină maximul dintre capetele din stânga ale intervalelor (notat cu r în secvența dată) și minimul dintre capetele din dreapta ale intervalelor (notat cu t în secvența dată). Intervalele se intersectează dacă $r \leq t$.

6. **Răspuns corect:** f) $a[n+1-j][n+1-i] \mid a[n+1-j, n+1-i]$

Indicații: Algoritmul parcurge elementele din zona **I** și, prin simetrie față de diagonală secundară, accesează elementele din zona **II**.

7. **Răspuns corect:** a) $p \neq 2 \mid p \text{ div } 2$

Indicații: Este folosit algoritmul căutării binare.

8. **Răspuns corect:** c) 3

Indicații: Variabila **S** de tip înregistrare, după toate atribuirile făcute, va avea valoarea {4, 5, 6, {3, 2, 1}}. Se afișează rezultatul expresiei: 5-2

9. **Răspuns corect:** a) 0

Indicații: Șirul apelurilor este: $f(6,2) \xRightarrow{6>2} f(2,6) \xRightarrow{2<6} 4$, apoi se revine la apelul anterior și se evaluează $f(4,3) \xRightarrow{4>3} f(3,4) \xRightarrow{3<4} 1$; se continuă cu $f(1,1) \xRightarrow{1=1} f(2,1) \xRightarrow{2>1} f(1,2) \xRightarrow{1<2} 1$ și se revine adunând pe 2 sau scăzând pe 1.

10. **Răspuns corect:** e) 8

Indicații: Circuitele elementare sunt: ABA, ACBA, ACDBA, ADBA, ADCBA, AEA, AEDBA, AEDCBA

11. **Răspuns corect:** b) **egr1 egrm**

Indicații: O abordare ar putea să plece de la o variantă dată. Folosind metoda backtracking, ținând cont de condițiile impuse de problemă, se pleacă de la primul cuvânt propus și se generează următoarele două soluții.

12. **Răspuns corect:** c) **$x=t[x]$; | $x:=t[x]$;**

Indicații: Se merge pe drumul de la nodul **x** către rădăcină, adică, de fiecare dată, se trece de la **x** la tatăl lui.

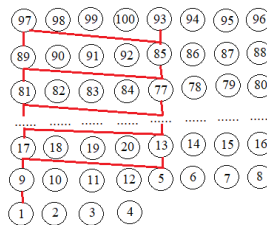
13. **Răspuns corect:** a) 1100

Indicații: Algoritmul determină cifra de control a numărului **n**. Deci, se cere să se determine câte numere din intervalul **[100, 10000]** au cifra de control **5**. Cum numerele cu aceeași cifră de control reprezintă o progresie aritmetică cu rația **9**, atunci răspunsul este $(10000-100)/9=1100$.

14. **Răspuns corect:** e) 148

Indicații: Grupăm elementele câte două (primul cu al doilea, al treilea cu al patrulea etc). Pentru prima pereche de elemente se face o comparare, presupunându-se că minimul perechii este minimul global, respectiv maximul perechii este maximul global. Pentru fiecare dintre celelalte perechi, se face o comparare ca să se determine cel mai mic și cel mai mare număr din pereche și apoi minimul perechii cu minimul global și maximul perechii cu maximul global. Deci, pentru fiecare pereche de elemente, se fac 3 comparații. Deci $3 \cdot \frac{100}{2} - 2 = 148$

15. **Răspuns corect:** d) 4



Indicații: Dacă aranjăm nodurile astfel: , se observă că o componentă conexă conține nodurile de pe două coloane. Adică, prima componentă conține nodurile de pe coloanele 1 și 5, a doua componentă conține nodurile de pe coloanele 2 și 6, a treia de pe coloanele 3 și 7 iar a patra de pe coloanele 4 și 8. Deci componentele conexe sunt:

{100, 96, 92, 88, 84, 80,, 12, 8, 4}
{99, 95, 91, 87, 83, 79,, 11, 7, 3}
{98, 94, 90, 86, 82, 78,, 10, 6, 2}
{97, 93, 89, 85, 81, 77,, 9, 5, 1}

