

Siruri de caractere

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);
```

Limbajul Pascal

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

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

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

Limbajul C++/ Limbajul C

```
char b[ ]="toc2019";
b[3]=b[3]-1;
strcpy(b+5,b+7);
strcpy(b,b+3);
```

Limbajul Pascal

```
var b: string[30];
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

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'));

Subprograme

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++	Limbajul C	Limbajul Pascal
<pre>void F(int &x, int y) { x=2; y=3; }</pre>	<pre>void F(int *x, int y) { *x=2; y=3; }</pre>	<pre>procedure F(var x: integer; y: integer); begin x:=2; y:=3; end;</pre>
a) 3 2 b) 1 2 c) 1 1 d) 2 3 e) 2 2 f) 3 3		

Tipul struct

- 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) $x.nr := x.nota$;

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) $(s.A.y+s.B.y)/2$

Recursivitate

- 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;
```

Page | 3

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

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');
```


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)
{ 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;
}
```

Limbajul Pascal

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

Page | 4

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

- 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

Backtracking

- 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:
- elementelor produsului cartezian A^5 , A fiind o mulțime cu 30 de elemente
 - partițiilor unei mulțimi
 - aranjamentelor de 30 de elemente luate câte 5
 - permutărilor de 5 elemente
 - combinărilor de 30 de elemente luate câte 5
 - submulțimilor de 5 elemente din mulțimea A, A fiind o mulțime cu 30 de elemente

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 |

Grafuri

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
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
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
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
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

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

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

a) $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$

b) $\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$

d) $\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

e) $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}$

f) $\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$