

**SUBIECTELE PROBEI PRACTICE PENTRU
EXAMENUL DE ATESTARE A COMPETENȚELOR PROFESIONALE A ABSOLVENȚILOR
CLASELOR DE MATEMATICĂ-INFORMATICĂ ȘI MATEMATICĂ-INFORMATICĂ,
INTENSIV INFORMATICĂ**

**PROGRAMARE
SPECIALIZAREA MATEMATICĂ-INFORMATICĂ**

Subiectul nr. 1:

- a) Subprogramul **cifre_pare** are ca parametru un număr natural **x** din intervalul $[0, 10^4]$. Subprogramul returnează suma cifrelor pare ale numărului **x** sau valoarea **-1**, dacă numărul nu conține nicio cifră pară. Scrieți la calculator definiția completă a subprogramului.
Exemplu: pentru **x=1234**, subprogramul returnează **6** iar pentru **x=9753**, subprogramul returnează **-1**.
- b) Fișierul **suma.in** conține pe prima linie un număr natural **n** ($1 \leq n \leq 100$), iar pe a doua linie un șir de **n** numere naturale din intervalul $[0, 10^4]$, separate prin câte un spațiu. Se cere scrierea și executarea la calculator a unui program care să calculeze suma cifrelor pare pentru fiecare număr de pe a doua linie a fișierului apoi să afișeze suma maximă calculată și primul număr care are această sumă, separate prin câte un spațiu. Programul va folosi apeluri utile ale subprogramului **cifre_pare**. Dacă nu există niciun număr care să conțină cifre pare, atunci pe ecran se afișează mesajul **doar cifre impare**.
Exemplu: dacă fișierul **suma.in** conține pe primul rând numărul **6**, iar pe al doilea rând numerele **2048 88 79 1848 1379 8606** se va afișa **20 1848**.

Subiectul nr. 2:

- a) Subprogramul **suma_x** are trei parametri, prin care primește:
- **n**, un număr natural din intervalul $[1, 100]$;
 - **a**, un tablou bidimensional cu **nxn** valori naturale;
 - **x**, un număr natural din intervalul $[1, n]$.
- Subprogramul returnează suma elementelor de pe linia **x**.
Liniile și coloanele tabloului sunt indexate de la **1**.
Scrieți la calculator definiția completă a subprogramului.
- b) Se consideră un graf neorientat **G** cu **n** noduri (**n** natural, $2 < n < 100$), dat prin matricea de adiacență **a**. Nodurile grafului sunt etichetate cu numere distincte de la **1** la **n**.
Fișierul **grade.in** conține mai multe linii. Pe prima linie a fișierului este scris un număr natural nenul **n** ($n < 100$), ce reprezintă numărul de noduri ale grafului, iar pe următoarele linii, perechi de numere naturale din intervalul $[1, 10^2]$ reprezentând nodurile între care există muchie. Se cere scrierea și executarea la calculator a unui program care citește din **grade.in** numărul **n**,

perechile de numere, construiește în memorie matricea de adiacență a grafului și afișează pe prima linie a ecranului, etichetele nodurilor cu grad maxim, iar pe a doua linie etichetele nodurilor cu grad 0 sau mesajul **nu exista**, în cazul în care nu există asemenea noduri. Pentru determinarea valorilor cerute se folosesc apeluri utile ale subprogramului **suma_x**.

Exemplu: dacă fișierul **grade.in** conține numerele

8

2 5

2 6

2 3

3 6

3 7

atunci se afișează, nu neapărat în această ordine:

2 3

1 4 8

Subiectul nr. 3:

a) Subprogramul **gaseste_cifra** are doi parametri:

- **x**, prin care primește un număr natural din intervalul $[0, 10^9]$;
- **c**, prin care primește o cifră.

Subprogramul returnează numărul de apariții ale cifrei **c** în scrierea numărului **x**. Scrieți la calculator definiția completă a subprogramului.

Exemplu: pentru **x=1323** și **c=3**, subprogramul returnează **2**; pentru **x=1323** și **c=4**, subprogramul returnează **0**.

b) Se citește de la tastatură un număr natural **nr** din intervalul $[0, 10^9]$. Se cere scrierea și executarea la calculator a unui program care să calculeze și să scrie, în fișierul **control.out**, valoarea formată din numărul de apariții ale fiecărei cifre în scrierea numărului **nr**, folosind apeluri utile ale subprogramului **gaseste_cifra**, iar pe linia următoare în fișier se va scrie cifra de control a lui **nr**. Cifra de control este cifra obținută prin însumarea cifrelor numărului și repetarea procedurii pentru suma obținută, până când se obține o singură cifră.

Exemplu: pentru **nr=23242527**, fișierul **control.out** va conține:

- pe primul rând, numărul **41110100** (pentru că cifrele **0** și **1** nu apar, **2** apare de **4** ori, **3** apare o dată, **4** apare o dată, **5** apare o dată, **6** nu apare, **7** apare o dată, **8** și **9** nu apar).
- pe al doilea rând, numărul **9** ($2+2+2+2+3+4+5+7=27$; $2+7=9$).

Subiectul nr. 4:

a) Subprogramul **puncte** are doi parametri:

- **cuv**, prin care primește o adresă cu ajutorul căreia memorează un șir de caractere reprezentând un text care conține cel puțin unul și cel mult **20** de caractere;
- **n**, prin care primește un număr natural din intervalul $[1, 20]$.

Subprogramul afișează pe ecran textul memorat începând cu adresa **cuv** urmat de **n** puncte.

Scrieți la calculator definiția completă a subprogramului.

Exemplu: dacă se memorează șirul de caractere **curat** cu ajutorul adresei **cuv** și **n=7**, pe ecran se afișează

curat.....

- b) Un text este format din cel puțin două cuvinte. Cuvintele din text sunt separate prin câte un spațiu. Textul are cel puțin **5** și cel mult **110** caractere reprezentând litere mici ale alfabetului englez, cratime și spații. Fiecare cuvânt din text are cel mult **20** de caractere.

Fișierul **cuvinte.in** conține pe prima linie un text de tipul precizat mai sus. Se cere scrierea și executarea la calculator a unui program care citește textul din fișier apoi afișează pe ecran, pe rânduri distincte, toate cuvintele din text urmate de un număr de puncte egal cu numărul de caractere al fiecărui cuvânt. Cuvintele se afișează în ordinea în care apar în text. Pentru rezolvare se folosesc apeluri utile ale subprogramului **puncte**. În același program se cere scrierea în fișierul **cuvinte.out** a acelor cuvinte din text, separate prin câte un spațiu, care conțin cel puțin o dată caracterul - (cratimă) și încep cu o consoană. Cuvintele se scriu în ordinea în care apar în text. Dacă în text nu există astfel de cuvinte, se scrie în fișierul **cuvinte.out** mesajul **nu exista**.

Exemplu: dacă fișierul **cuvinte.in** conține textul

cine si-a pierdut o zi cat o viata s-o caute repede

atunci pe ecran se afișează

cine....

si-a....

pierdut.....

o.

zi..

cat...

o.

viata.....

s-o...

caute.....

repede.....

iar fișierul **cuvinte.out** conține

si-a s-o

Subiectul nr. 5:

- a) Subprogramul **nrpare_p** are doi parametri, prin care primește:

- **n**, un număr natural din intervalul **[5, 100]**;
- **v**, un tablou bidimensional cu **n** linii și **n** coloane de numere naturale din intervalul **[0, 10³]**;

Subprogramul returnează numărul elementelor pare situate pe diagonala principală a tabloului **v** sau valoarea -1 dacă diagonala nu conține valori pare.

Subprogramul **nrpare_s** are doi parametri, prin care primește:

- **n**, un număr natural din intervalul **[5, 100]**;
- **v**, un tablou bidimensional cu **n** linii și **n** coloane de numere naturale din intervalul **[0, 10³]**;

Subprogramul returnează numărul elementelor pare situate pe diagonala secundară a tabloului **v** sau **-1** dacă diagonala nu conține valori pare.

Scrieți la calculator definițiile complete ale celor două subprograme.

- b) Fișierul **matrice.in** conține pe prima linie un număr natural **par n** ($2 \leq n \leq 100$), iar pe următoarele **n** linii câte **n** numere naturale din intervalul $[0, 10^3]$, separate prin câte un spațiu. Se cere scrierea și executarea la calculator a unui program care să citească din fișier **n** apoi cele **n · n** elemente ale tabloului și să afișeze pe ecran numărul valorilor impare situate pe cele două diagonale ale tabloului. Pentru afișarea valorii cerute se folosesc apeluri utile ale subprogramelor **nrpare_p** și **nrpare_s**. Dacă nu există nicio valoare impară pe diagonale atunci pe ecran se afișează mesajul **nu exista**.

Exemplu: dacă fișierul **matrice.in** conține numerele:

6

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 6 7 8 9 0

1 2 3 4 5 6

atunci se afișează 6.

Subiectul nr. 6:

- a) Subprogramul **linie_x** are trei parametri, prin care primește:

- **n**, un număr natural din intervalul $[1, 100]$;
- **a**, un tablou bidimensional cu **n × n** valori naturale;
- **x**, un număr natural din intervalul $[1, n]$.

Subprogramul returnează numărul elementelor nenule de pe linia **x**.

Liniile și coloanele tabloului sunt indexate de la 1.

Scrieți la calculator definiția completă a subprogramului.

- b) Subprogramul **coloana_x** are trei parametri, prin care primește:

- **n**, un număr natural din intervalul $[1, 100]$;
- **a**, un tablou bidimensional cu **n × n** valori naturale;
- **x**, un număr natural din intervalul $[1, n]$.

Subprogramul returnează numărul elementelor nenule de pe coloana **x**.

Scrieți la calculator definiția completă a subprogramului.

- c) Se consideră un graf orientat **G** cu **n** vârfuri (**n** natural, $2 < n < 100$), dat prin matricea de adiacență **a**. Vârful grafului sunt etichetate cu numere distincte de la 1 la **n**.

Fișierul **graf.in** conține mai multe linii. Pe prima linie a fișierului este scris un număr natural nenul **n** ($n < 100$), ce reprezintă numărul de vârfuri ale grafului, iar pe următoarele linii, perechi de numere naturale din intervalul $[1, 10^2)$ reprezentând vârfurile între care există arc. Se cere scrierea și executarea la calculator a unui program care citește din fișierul **graf.in** numărul **n**, perechile de numere, construiește în memorie matricea de adiacență a grafului și afișează pe prima

linie a ecranului, separate prin câte un spațiu, vârfurile care au gradul interior egal cu gradul exterior sau mesajul **nu exista** pentru cazul în care nu există vârfuri cu această proprietate, iar pe a doua linie a ecranului, numărul de arce ale grafului. Pentru afișarea valorilor cerute se folosesc apeluri utile ale subprogramelor **linie_x** și **coloana_x**.

Exemplu: dacă fișierul **graf.in** conține numerele

6

1 2

3 1

3 2

4 3

5 3

6 5

atunci se afișează, nu neapărat în această ordine:

1 5 3

6

Subiectul nr. 7:

Se consideră șirul 0, 1, 2, 0, 1, 2, 0,... obținut prin scrierea secvențelor de numere naturale mai mici decât 3, în cadrul secvenței numerele fiind scrise în ordine strict crescătoare. Se cere scrierea și executarea la calculator a unui program care citește de la tastatură un număr natural n ($2 \leq n \leq 40$) și construiește în memorie un tablou bidimensional cu n linii numerotate de la 0 la $n-1$ și n coloane numerotate de la 0 la $n-1$, astfel încât prin parcurgerea liniilor cu indice par de la stânga la dreapta și prin parcurgerea liniilor cu indice impar de la dreapta la stânga, se obțin primele n numere ale șirului descris mai sus. Tabloul astfel construit va fi scris în fișierul **mat.out**, fiecare linie a tabloului pe câte o linie a fișierului și elementele fiecărei linii separate prin câte un spațiu.

Exemplu: dacă pentru n se citește 5, în fișierul **mat.out** va avea următorul conținut:

0 1 2 0 1

1 0 2 1 0

0 1 2 0 1

1 0 2 1 0

0 1 2 0 1

Subiectul nr. 8:

- a) Subprogramul **nr_prim** are un singur parametru n prin care primește un număr natural cu cel mult patru cifre și returnează 1 dacă n este prim sau 0 altfel.

Subprogramul **cmmdc** are doi parametri prin care primește câte un număr natural nenul cu cel mult patru cifre. Subprogramul returnează cel mai mare divizor comun al celor două numere.

Scrieți la calculator definițiile complete ale celor două subprograme.

- b) Fișierul **numere.in** conține pe prima linie cel puțin două și cel mult 10^6 numere naturale din intervalul $[1, 10^4]$, separate prin câte un spațiu. Se cere scrierea și executarea la calculator a unui program care să afișeze pe ecran numărul perechilor de numere citite consecutiv din fișier care nu sunt numere prime, dar sunt numere prime între ele.

Exemplu: dacă fișierul **numere.in** conține numerele:

35 45 22 11 8 35 256 1000

atunci se afișează **3**

Explicație: Se numără perechile **(45, 22)**, **(8, 35)**, **(35, 256)**

Subiectul nr. 9:

- a) Oglinditul unui număr este numărul scris cu cifrele primului număr, în ordine inversă. Subprogramul **oglinda** primește ca parametru un număr natural **x** din intervalul $[1, 10^6]$ și returnează oglinditul acestuia. Scrieți la calculator definiția completă a subprogramului.

Exemplu: pentru **x=12401**, subprogramul returnează numărul **10421**.

- b) Se citesc de la tastatură două numere naturale **a** și **b** din intervalul $[1, 10^6]$. Un număr este palindrom dacă este egal cu oglinditul său. Se cere scrierea și executarea la calculator a unui program care scrie în fișierul **palindrom.out**, separate prin câte un spațiu, în ordine crescătoare, toate numerele din intervalul $[a, b]$ care sunt palindromuri. Apoi, programul citește de la tastatură un număr natural **k** din intervalul $[1, 10^7]$ și scrie, pe următorul rând al fișierului **palindrom.out**, palindromul asociat lui **k**. Palindromul asociat unui număr este cel mai mic palindrom mai mare decât numărul inițial. Dacă numărul **k** este palindrom, se va scrie mesajul **palindrom**.

Pentru determinarea valorilor cerute, programul va folosi apeluri utile ale subprogramului **oglinda**.

Exemplu:

- pentru **a=11** și **b=130**, fișierul **palindrom.out** va conține pe primul rând numerele **11 22 33 44 55 66 77 88 99 101 111 121**
- pentru **k=39995**, fișierul **palindrom.out** va conține pe al doilea rând numărul **40004**

Subiectul nr. 10:

Din fișierul **sir.in** se citesc un număr natural nenul **n** ($n < 1000$) și un șir de **n** numere întregi cu cel mult 5 cifre fiecare.

Se cere scrierea și executarea la calculator a unui program care scrie în fișierul **sir.out**, separate prin câte un spațiu, în ordine crescătoare, toate numerele naturale de 2 cifre care nu apar în fișierul **sir.in**. În cazul în care nu există astfel de numere, în fișier se va scrie mesajul **nu exista**.

Exemplu:

Fișierul **sir.in** conține

30

În fișierul **sir.out** se va scrie

10 12 14 15 16 17 18 20 23 24
25 26 28 29 30 31 32 33 34 36
38 39 40 41 43 45 46 47 48 49

3 84 2 6 21 44 21 55 4 -7 55 37
74 68 85 5 42 67 13 19 27 11 99
87 57 1 94 63 35 22

50 51 52 53 54 56 58 59 60 61
62 64 65 66 69 70 71 72 73 75
76 77 78 79 80 81 82 83 86 88
89 90 91 92 93 95 96 97 98

Subiectul nr. 11:

a) Subprogramul **vocale** are doi parametri:

- **s**, prin care primește o adresă cu ajutorul căreia memorează un șir de caractere reprezentând un text care conține cel puțin unul și cel mult **80** de caractere;
- **v**, prin care furnizează un număr natural reprezentând numărul de vocale (litere mici sau litere mari din mulțimea { **a, e, i, o, u, A, E, I, O, U** }) din șirul de caractere memorat cu ajutorul variabilei **s**.

Scrieți la calculator definiția completă a subprogramului.

Exemplu: dacă se memorează, cu ajutorul variabilei **s**, șirul de caractere **Cum va place** variabila **v** va furniza valoarea **4** (deoarece șirul de caractere conține patru vocale).

b) Un text este format din cel puțin două cuvinte. Cuvintele din text sunt separate prin câte un spațiu. Textul are cel puțin **3** și cel mult **80** de caractere reprezentând litere mici ale alfabetului englez, litere mari ale alfabetului englez și spații.

Fișierul **vocale.in** conține pe prima linie un text de tipul precizat mai sus. Se cere scrierea și executarea la calculator a unui program care citește textul din fișier, apoi afișează pe ecran numărul cuvintelor din text care conțin numai vocale (litere mici sau litere mari din mulțimea {**a, e, i, o, u, A, E, I, O, U**}). Pentru rezolvare se folosesc apeluri utile ale subprogramului **vocale**. În același program se cere scrierea cuvintelor din text care conțin cel puțin o dată șirul de caractere **str**. Cuvintele se scriu în fișierul **vocale.out** pe rânduri diferite, în ordinea în care apar în text. Dacă în text nu există astfel de cuvinte, se scrie în fișierul **vocale.out** mesajul **nu exista**.

Exemplu: dacă fișierul **vocale.in** conține textul
Au o gradina si o fereastră albastra

atunci pe ecran se afișează

3

deoarece sunt trei cuvinte care conțin numai vocale

iar fișierul **vocale.out** conține

fereastră

albastra

Subiectul nr. 12:

a) Un număr perfect este un număr natural egal cu dublul sumei divizorilor săi.

Exemplu: **6** este număr perfect pentru că $2 \cdot 6 = 1 + 2 + 3 + 6$.

Subprogramul **perfect** primește prin parametrul **x** un număr natural din intervalul $[1, 10^6]$ și returnează **1**, dacă numărul este perfect și **0**, altfel. Scrieți la calculator definiția completă a subprogramului.

Exemplu: pentru **x=6** subprogramul returnează **1**, iar pentru **x=5** subprogramul returnează **0**.

- b) Fișierul **perfect.in** conține pe primul rând un șir de numere naturale din intervalul $[1, 10^6]$, separate prin câte un spațiu. Se cere scrierea și executarea la calculator a unui program care citește șirul de numere din fișier, apoi afișează pe ecran cel mai mare număr perfect din șirul citit și prima poziție a acestuia în șir, pe același rând, separate prin câte un spațiu. Programul va folosi apeluri utile ale subprogramului **perfect**. Dacă în fișier nu există niciun număr perfect, atunci pe ecran se afișează mesajul **imposibil**.

Exemplu: dacă fișierul **perfect.in** conține numerele 2048 6 79 496 1379 28 se va afișa 496 4

Subiectul nr. 13:

- a) Subprogramul **pozitie** are trei parametri prin care primește:

- **n**, un număr natural din intervalul $[2, 100]$;
- **v**, un tablou unidimensional cu cel mult **n** numere naturale din intervalul $[0, 10^2]$;
- **x**, un număr natural din intervalul $[1, n]$.

Subprogramul returnează poziția primei apariții a numărului **x** în tabloul **v** sau **-1** în cazul în care numărul **x** nu apare în tablou.

Tabloul unidimensional este indexat de la 1.

Scrieți la calculator definiția completă a subprogramului.

Exemplu: pentru **n=7**, tabloul **v=(5, 1, 0, 8, 0, 2, 3)** și **x=0**, se returnează **3**.

- b) Se consideră un arbore cu rădăcină cu **n** noduri, dat prin vectorul “de tați”. Nodurile sunt etichetate cu numere distincte de la 1 la **n**.

Fișierul **arbore.in** conține două linii. Pe prima linie a fișierului este scris un număr natural nenul **n** ($n < 100$) ce reprezintă numărul de noduri ale arborelui, iar pe următoarea linie sunt scrise **n** numerele naturale din intervalul $[1, 10^2]$, reprezentând valorile componentelor vectorului “de tați”. Se cere scrierea și executarea la calculator a unui program care citește din fișierul **arbore.in** numărul **n** și cele **n** elemente ale vectorului “de tați” și afișează pe ecran, pe prima linie eticheta nodului rădăcină, pe a doua linie nodurile cu cei mai mulți fii, iar pe a treia linie etichetele nodurilor de tip frunză. Valorile se afișează separate prin câte un spațiu, în ordine crescătoare. În rezolvare se folosesc apeluri utile ale subprogramului **pozitie**.

Exemplu: dacă fișierul **arbore.in** conține numerele

10

2 0 2 2 3 3 5 3 1 1

atunci se afișează:

2

2 3

4 6 7 8 9 10

Subiectul nr. 14:

- a) Subprogramul **secv_tablou** are patru parametri, prin care primește :

- **n**, un număr natural din intervalul **[5, 100]**;
- **v**, un tablou unidimensional cu cel mult **n** numere naturale din intervalul **[1, 10³]**;
- **a** și **b**, câte un număr natural din intervalul **[1, n]**, (**a < b**).

Subprogramul afișează pe ecran, separate prin câte un spațiu, elementele din tablou care se află între pozițiile **a** și **b**. Pozițiile elementelor în tablou sunt numerotate de la 1.

Scrieți la calculator definiția completă a subprogramului.

Exemplu: pentru **n=7**, tabloul **v=(5, 1, 3, 8, 7, 2, 3)**, **a=2**, **b=5**, se afișează **1 3 8 7**.

- b) O secvență dintr-un șir de valori este numită **secventa_k** dacă începe și se termină cu aceeași valoare **k** și conține această valoare de cel puțin două ori.

Exemplu: **3, 8, 7, 2, 3** este **secventa_k** a tabloului **(5, 1, 3, 8, 7, 2, 3)**.

Fișierul **secventa_k.in** conține pe prima linie un număr natural nenul **n** (**n ≤ 100**), iar pe linia a doua **n** numere naturale din intervalul **[1, 10³]**, separate prin câte un spațiu. Se cere scrierea și executarea la calculator a unui program care să citească de la tastatură o valoare **k** (**1 ≤ k ≤ 10³**), apoi din fișierul **secventa_k.in** valoarea lui **n** și cele **n** valori de pe a doua linie a fișierului și care să afișeze, pe rânduri distincte, toate secvențele de forma **secventa_k**. Numerele din secvență se afișează separate prin câte un spațiu. Pentru afișarea secvențelor cerute se folosesc apeluri utile ale subprogramului **secv_tablou**. Dacă nu există nicio secvență de forma **secventa_k** atunci pe ecran se afișează mesajul **imposibil**.

Exemplu: dacă fișierul **secventa_k.in** are următorul conținut

10

3 5 1 5 2 3 5 2 5 1

iar de la tastatură se citește pentru **k** valoarea 5

atunci se afișează, nu neapărat în această ordine:

5 1 5

5 1 5 2 3 5

5 1 5 2 3 5 2 5

5 2 3 5

5 2 3 5 2 5

5 2 5

Subiectul nr. 15:

- a) Subprogramul **consoane** are un parametru, **s**, prin care primește o adresă cu ajutorul căreia memorează un șir de caractere reprezentând un text care conține cel puțin unul și cel mult 90 de caractere și returnează un număr natural reprezentând numărul de consoane (litere mici sau litere mari care nu aparțin mulțimii {**a, e, i, o, u, A, E, I, O, U**}) din șirul de caractere memorat cu ajutorul variabilei **s**. Scrieți la calculator definiția completă a subprogramului.

Exemplu: dacă se memorează, cu ajutorul variabilei **s**, șirul de caractere **Ce mai primavara** subprogramul **consoane** va returna valoarea 7 (deoarece șirul de caractere conține 7 consoane)

- b) Un text este format din cel puțin două cuvinte. Cuvintele din text sunt separate prin câte un spațiu. Textul are cel puțin 3 și cel mult 90 de caractere reprezentând litere mici ale alfabetului englez, litere mari ale alfabetului englez și spații.

Fișierul **consoane.in** conține pe prima linie un text de tipul precizat mai sus. Se cere scrierea și executarea la calculator a unui program care citește din fișier textul și afișează pe ecran cel mai mare număr de consoane ale unui cuvânt din text (consoanele se consideră literele mici sau literele mari care nu aparțin mulțimii {**a, e, i, o, u, A, E, I, O, U**}). Pentru rezolvare se folosesc apeluri utile ale subprogramului **consoane**. În același program se cere scrierea în fișierul **consoane.out**, pe rânduri diferite, a cuvintelor din text care au un număr par de caractere și încep cu literă mare. Cuvintele se scriu în ordinea în care apar în text. Dacă în text nu există astfel de cuvinte, se scrie în fișierul **consoane.out** mesajul **nu exista**.

Exemplu: dacă fișierul **consoane.in** conține textul

Mult Zgomot pentru Nimic

atunci pe ecran se afișează

4

deoarece în text cuvintele **Zgomot** și **pentru** au număr maxim de consoane (**4**),

iar fișierul **consoane.out** conține

Mult

Zgomot

Subiectul nr. 16:

- a) Scrieți la calculator definiția completă a subprogramului **nr_div** care primește prin parametrul **a** un număr natural nenul cu cel mult patru cifre și returnează numărul divizorilor proprii ai numărului **a**.
- b) Fișierul **divizori.in** conține cel mult 10^6 numere naturale din intervalul $[1, 10^4]$ separate prin câte un spațiu. Se cere scrierea și executarea la calculator a unui program care citește numerele din fișier, apoi afișează pe ecran numărul din fișier cu număr maxim de divizori. Dacă există mai multe numere cu număr maxim de divizori, se va afișa ultimul dintre acestea.

Exemplu: dacă fișierul **divizori.in** conține numerele:

13 36 45 22 7 8 100 32

atunci se afișează **100**

Explicație: Numerele cu cei mai mulți divizori sunt **36** și **100** dar, dintre acestea, ultimul în fișier este **100**.

Subiectul nr. 17:

Se consideră tipul de date **clipa**, definit astfel:

```
struct clipa
```

```
{
```

```
    int sec, mi, ora;
```

```
};
```

unde valorile memorate în câmpurile **sec** și **mi** sunt numere naturale din intervalul $[0, 59]$, iar valoarea memorată în câmpul **ora** este un număr natural din intervalul $[0, 23]$.

- a) Subprogramul **timp** primește ca parametru variabila **c** de tip **clipa** și afișează pe ecran conținutul acesteia în format **hh:mm:ss**. Scrieți la calculator definiția completă a subprogramului.

Exemplu: pentru **c.sec=35**, **c.min=4** și **c.ora=6**, se va afișa **06:04:35**.

- b) Fișierul **clipa.in** conține pe primul rând un număr natural **n** ($1 \leq n \leq 20$), iar pe următoarele **n** rânduri, câte trei valori, corespunzătoare câmpurilor structurii **clipa**, separate prin câte un spațiu. Se cere scrierea și executarea la calculator a unui program care citește datele din fișier, apoi afișează pe ecran, unul sub altul, momentele citite, în format **hh:mm:ss**. Programul afișează apoi, pe rândul următor, momentul cel mai apropiat de sfârșitul zilei, în format **hh:mm:ss**. Programul va folosi apeluri utile ale subprogramului **timp**.

Exemplu: dacă fișierul **clipa.in** are următorul conținut

```
5
20 45 6
1 2 3
4 56 3
59 59 23
34 3 4
```

pe ecran se va afișa:

```
06:45:20
03:02:01
03:56:04
23:59:59
04:03:34
23:59:59
```

Subiectul nr. 18:

Se consideră tipul de date **punct** definit astfel

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

unde **x** și **y** reprezintă coordonatele carteziene ale unui punct.

Subprogramul **distanța** are doi parametri **A** și **B** de tip **punct** prin care primește coordonatele punctelor **A** și **B** în plan.

Subprogramul returnează distanța dintre punctele **A** și **B**.

Scrieți la calculator definiția completă a subprogramului.

- a) Fișierul **puncte.in** conține pe prima linie numărul real nenul **r** și numărul natural **n** ($2 < n < 100$). Pe fiecare dintre următoarele **n** linii este scrisă câte o pereche de numere reale **x** și **y**, reprezentând coordonatele carteziene ale unui punct în planul **xOy**.

Se cere scrierea și executarea la calculator a unui program care citește din fișierul **puncte.in** numerele **r**, **n**, apoi cele **n** perechi de numere reale și afișează pe ecran coordonatele punctelor aflate pe cercul cu centrul în originea sistemului de coordonate și raza **r** sau mesajul **nu exista** în cazul în care niciunul dintre puncte nu îndeplinește această condiție. Valorile se afișează pe rânduri diferite, în format (**abscisă**, **ordonată**). Pentru afișarea valorilor cerute se folosesc apeluri utile ale subprogramului **distanța**.

Exemplu: dacă fișierul **puncte.in** conține numerele

```
5
6
0 2
```

5 0
3 2
4 3
5 3
0 -5

atunci se afișează:

(5,0)
(4,3)
(0,-5)

Subiectul nr. 19:

Fișierul **numere.in** conține cel puțin două și cel mult 10^4 numere întregi cu cel mult 5 cifre fiecare.

Se cere scrierea și executarea la calculator a unui program care citește valorile din fișier și scrie în fișierul **numere.out**, cele mai mari două numere din șir și de câte ori apar acestea, astfel: pe prima linie din fișier cel mai mare număr urmat de numărul său de apariții iar pe a doua linie celălalt număr urmat de numărul său de apariții. Valorile de pe aceeași linie se scriu separate prin câte un spațiu. Dacă în fișierul **numere.in** nu există cel puțin două numere distincte, în fișierul **numere.out** se scrie mesajul **numere identice**.

Exemplu :

Fișierul **numere.in** conține
3 4 9 -2 -7 7 9 9 4 1

În fișierul **numere.out** se va afișa
9 3
7 1

Subiectul nr. 20:

- a) Subprogramul **cifre** are un parametru, **s**, prin care primește o adresă cu ajutorul căreia memorează un șir de caractere având cel puțin unul și cel mult 100 de caractere și returnează un număr natural reprezentând numărul de cifre din șirul de caractere memorat cu ajutorul variabilei **s**. Scrieți la calculator definiția completă a subprogramului.

Exemplu: dacă se memorează, cu ajutorul variabilei **s**, șirul de caractere **11c** subprogramul **cifre** va returna valoarea 2 (deoarece șirul de caractere conține 2 cifre)

- b) Un șir de caractere are cel puțin 5 și cel mult 100 de caractere reprezentând litere mici ale alfabetului englez, cifre, virgule sau spații. Șirul de caractere este format din cel puțin două entități lexicale separate între ele prin spații sau virgule. Entitățile lexicale sunt formate din litere mici ale alfabetului englez sau cifre. Șirul conține cel puțin o entitate lexicală care începe cu o literă mică. Fișierul **cifra.in** conține pe prima linie un șir de caractere de tipul precizat mai sus. Se cere scrierea și executarea la calculator a unui program care citește șirul de caractere din fișier, apoi afișează pe ecran, în ordinea în care apar, separate prin câte un spațiu, numai entitățile lexicale din șir care conțin cel puțin o cifră. Dacă în șirul de caractere nu există astfel de entități lexicale, se afișează pe ecran mesajul **nu exista**. Pentru rezolvare se folosesc apeluri utile ale

subprogramului **cifre**. În același program se cere transformarea în literă mare a tuturor inițialelor entităților lexicale din șirul de caractere care încep cu o literă mică. Entitățile lexicale care încep cu o cifră rămân nemodificate. După modificare, entitățile lexicale se scriu în fișierul **cifra.out** pe rânduri diferite, în ordinea în care apar în șir.

Exemplu: dacă fișierul **cifra.in** conține șirul de caractere

vrei pana la 1000, poti pana la 6

atunci pe ecran se afișează

Vrei

Pana

La

1000

Poti

Pana

La

6

iar fișierul **cifra.out** conține

1000 6

Subiectul nr. 21:

- Scrieți la calculator definiția completă a subprogramului **nr_div** care primește prin parametrul **a** un număr natural nenul cu cel mult patru cifre și returnează numărul tuturor divizorilor lui **a**.
- Scrieți și executați la calculator un program care citește de la tastatură un șir de numere naturale nenule care se termină cu numărul **1**. Șirul are cel puțin unul și cel mult 10^2 termeni cu cel mult patru cifre fiecare. Programul verifică, apelând subprogramul **nr_div**, dacă numerele citite sunt pătrate perfecte ale unui număr prim. Rădăcinile pătrate ale acestor numere se scriu în fișierul **patrate.out**, separate prin câte un spațiu, în ordinea din șir. Dacă niciun număr citit nu este pătratul unui număr prim, atunci în fișier se scrie mesajul **nu exista**.

Exemplu: dacă se citesc numerele **13 56 169 196 7 121 100 49 1**

atunci fișierul **patrate.out** conține numerele:

13 11 7

Explicație: un număr care este pătratul unui număr prim are 3 divizori.

Subiectul nr. 22:

- Subprogramul **tablou** are doi parametri:
 - n**, prin care furnizează un număr natural din intervalul **[2, 100]**;
 - v**, prin care furnizează un tablou unidimensional cu **n** numere întregi din intervalul **$[-10^4, 10^4]$** ;

Subprogramul construiește tabloul **v** cu **n** numere citite de la tastatură.

Scrieți la calculator definiția completă a subprogramului.

- b) Se cere scrierea și executarea la calculator a unui program care, folosind apeluri utile ale subprogramului **tablou**, construiește un tablou unidimensional **a** cu **na** elemente *ordonate strict crescător*, apoi construiește un tablou unidimensional **b** cu **nb** elemente *ordonate strict descrescător*. Programul va scrie în fișierul **comune.out**, în *ordine crescătoare*, separate prin câte un spațiu, elementele comune celor două tablouri sau mesajul **nu exista** dacă nu sunt astfel de elemente.

Exemplu: dacă se citesc de la tastatură numerele

5

1 2 3 4 5

6

9 7 5 3 2 0

atunci fișierul **comune.out** va conține:

2 3 5

Subiectul nr. 23:

- a) Subprogramul **prim** primește prin intermediul parametrului **a** un număr natural cu cel mult 5 cifre și returnează **1** dacă **a** este număr prim și **0** în caz contrar. Scrieți la calculator definiția completă a subprogramului.
- b) Fișierul **numere.in** conține un șir cu cel puțin unul și cel mult 10^6 numere naturale cu cel mult 5 cifre fiecare. Se cere scrierea și executarea la calculator a unui program care citește din fișierul **numere.in** șirul de valori și scrie în fișierul **numere.out** numărul de elemente conținut de cea mai lungă secvență de numere prime aflate pe poziții consecutive în șirul citit. Dacă există mai multe secvențe cu același număr maxim de elemente, în fișierul **numere.out** se va scrie numărul acestor secvențe.

Exemplu1:

Fișierul **numere.in** conține

3 5 4 9 7 5 11 5 4 1 3 2 3 4

În fișierul **numere.out** se va scrie

4

Exemplu2:

Fișierul **numere.in** conține

3 5 4 9 7 5 11 5 4 1 3 2 5 11 4

În fișierul **numere.out** se va scrie

2

Subiectul nr. 24:

- a) Subprogramul **numar_cifre** are un parametru, **x**, prin care primește un număr natural din intervalul $[0, 10^8)$. Subprogramul returnează numărul de cifre ale lui **x**.
Scrieți la calculator definiția completă a subprogramului.
Exemplu: pentru **x=207**, subprogramul returnează 3.
- b) Un număr natural **a** este numit **extrem** al unui număr natural **b** dacă are același număr de cifre cu **b** și, în plus, este maxim în cazul în care **b** este par și este minim în cazul în care **b** este impar.
Exemplu: 999 este **extrem** al numărului 514, iar 100 este **extrem** al numărului 517.

Fișierul **extrem.in** conține cel mult **1000** de numere naturale din intervalul $[0, 10^8)$, separate prin câte un spațiu. Se cere scrierea și executarea la calculator a unui program care citește numerele din fișier, apoi afișează pe ecran numerele din fișier, respectiv numerele de tip extrem corespunzătoare numerelor din fișier, în ordinea în care acestea apar în fișier. Pentru determinarea numerelor cerute se folosesc apeluri utile ale subprogramului **numar_cifre**. Numerele se afișează pe rânduri diferite, în formatul **b . . a**, unde **b** este numărul par din fișier iar **a** este numărul extrem corespunzător lui **b** sau în formatul **a . . b**, unde **b** este numărul impar din fișier iar **a** este numărul extrem corespunzător lui **b**.

Exemplu: dacă fișierul **extrem.in** conține numerele **2153 532 1504 61 7**

atunci pe ecran se afișează

1000 . . 2153

532 . . 999

1504 . . 9999

10 . . 61

0 . . 7

Subiectul nr. 25:

- a) Subprogramul **majuscule** are un parametru, **s**, prin care primește o adresă cu ajutorul căreia memorează un text format din cel puțin **3** și cel mult **50** de caractere.

Subprogramul returnează numărul de litere mari din textul memorat cu ajutorul variabilei **s**.

Scrieți la calculator definiția completă a subprogramului.

Exemplu: dacă se memorează, cu ajutorul variabilei **s**, textul **DON QUIJOTE de la Mancha**, subprogramul returnează **11**.

- b) Un text este format din titluri de cărți separate între ele prin caracterul ***** (asterisc). Textul are cel mult **200** de caractere și este format din cel puțin două titluri de cărți. Fiecare titlu are cel puțin **3** și cel mult **50** de caractere reprezentând litere mici ale alfabetului englez, litere mari ale alfabetului englez, virgule, puncte sau spații.

Fișierul **titlu.in** conține pe prima linie un text de tipul precizat mai sus. Se cere scrierea și executarea la calculator a unui program care citește textul din fișier, apoi afișează pe ecran numărul de titluri din text care conțin exact două litere mari. Pentru rezolvare se folosesc apeluri utile ale subprogramului **majuscule**. În același program se cere scrierea în fișierul **titlu.out**, pe rânduri diferite, respectând ordinea în care apar în text, a fiecărui titlu în care apare cel puțin o dată litera **M**. Dacă în text nu există niciun titlu cu această proprietate, în fișierul **titlu.out** se scrie mesajul **nu exista**.

Exemplu: dacă fișierul **titlu.in** conține textul

Muntele vrajit*Fratii Karamazov*DON QUIJOTE de la Mancha

atunci pe ecran se afișează

1

deoarece în text numai titlul **Fratii Karamazov** are exact două litere mari

iar fișierul **titlu.out** conține

Muntele vrajit



DON QUIJOTE de la Mancha

deoarece acestea sunt titlurile în care apare cel puțin o dată litera **M**.

Colectivul de autori:

Săcuiu Silviu Eugen	- Colegiul Național „Mihai Viteazul”
Anca Mihaela	- Colegiul Național „Ion Luca Caragiale”
Florea Andrei	- Colegiul Național „Ion Luca Caragiale”
Balcă Mariana Mihaela	- Colegiul Național ”Școala Centrală”
Matei Elena Raluca	- Liceul Teoretic „Jean Monnet”
Ispășescu Smaranda	- Liceul Teoretic Bulgar „Hristo Botev”

Coordonator:

Ștefania Penea – inspector școlar pentru Informatică și Tehnologia Informației și a Comunicațiilor,

Inspectoratul Școlar al Municipiului București