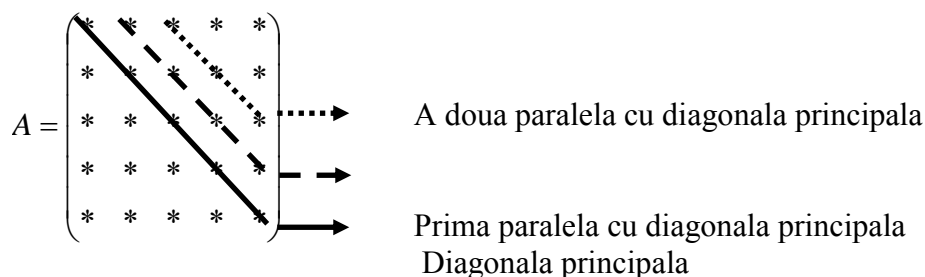


**VIZAT,  
INSPECTOR ȘCOLAR GENERAL,  
PROF. GHERGHE VALERIA**

**SUBIECTELE PROBEI PRACTICE PENTRU  
EXAMENUL DE ATESTAT PROFESIONAL LA INFORMATICĂ, 2014  
PROGRAMARE INTENSIV**

1. Se considera o matrice  $A$  cu  $m$  linii și  $n$  coloane ce conține numere întregi. Să se scrie un program Pascal care afișează, pe o singură linie în fișierul text  $A.txt$ , elementele situate pe conturul matricei. Se porneste din poziția  $(1,1)$  și parcurgerea se va face în sensul acelor de ceasornic.
2. Se considera o matrice  $A$  cu  $m$  linii și  $n$  coloane de numere reale și un vector  $V$  cu  $m$  elemente (numere reale). Să se verifice dacă elementele vectorului  $V$  formează o coloană a matricei  $A$ . În caz afirmativ să se afișeze în fișierul  $verific.txt$  numărul coloanei respective; în caz contrar să se afișeze în fișier 'NU'.
3. Se considera o matrice  $A$  cu  $m$  linii și  $n$  coloane ce conține numere reale. Să se ștergă din matrice linia a 3-a. Să se afișeze în fișierul  $A.txt$  matricea obținută în format corespunzător. Se considera că matricea are cel puțin 3 linii ( $m \geq 3$ ).
4. Se considera o matrice  $A$  cu  $m$  linii și  $n$  coloane de numere întregi. Să se introducă o nouă linie la începutul matricii. Să se afișeze matricea obținută în format corespunzător în fișierul  $A.txt$ .
5. Se considera o matrice pătratică de dimensiune  $n(n > 2)$  ce conține numere întregi. Să se calculeze media aritmetică a elementelor situate pe a doua paralelă cu diagonala principală situată deasupra diagonalei principale. Valoarea se va afișa în fișierul  $B.txt$ .



6. Fie șirul următor definit prin recurență:

$$a_n = \begin{cases} -n + a_{n-3}, & \text{daca } n \text{ este impar;} \\ 2n + a_{n-3}, & \text{daca } n \text{ este par,} \end{cases}$$

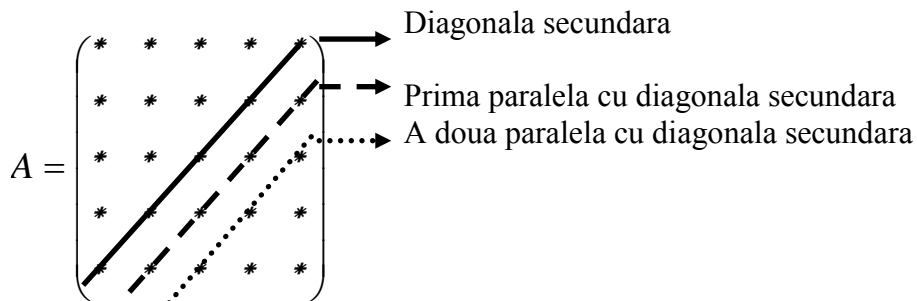
$$a_0 = 0,$$

$$a_1 = 1,$$

$$a_2 = 2.$$

Sa se scrie un subprogram recursiv care calculeaza termenii acestui sir si sa se foloseasca acest subprogram pentru a calcula al 10-lea termen al sirului. Dimensiunea vectorului si elementele acestuia se gasesc pe o linie in fisierul A.TXT.

7. Sa se scrie un subprogram recursiv care afiseaza elementele situate pe pozitii pare intr-un vector ce contine numere intregi. Elementele vor fi afisate fiecare pe cate un rand si vor fi afisate in ordinea in care apar in vector. Dimensiunea vectorului si elementele acestuia se gasesc pe o linie in fisierul C.TXT
8. Fie A o matrice patratica de dimensiune  $n(n > 2)$  de numere reale. Cele doua diagonale impart matricea in 4 regiuni (4 triunghiuri). Sa se verifice daca suma elementelor din interiorul triunghiului de sus este egala cu suma elementelor din interiorul triunghiului de jos. Dimensiunea matricei si elementele acesteia se gasesc in fisierul INTRARE.txt astfel: pe prima linie se gaseste n, pe urmatoarele n linii se gaseste cate o linie din matrice.
9. Se considera o matrice patratica de dimensiune  $n(n > 2)$  ce contine numere intregi. Sa se afiseze in fisierul A.TXT suma elementelor situate pe prima paralela cu diagonala secundara situata sub diagonala secundara.



10. Se considera o matrice cu m linii si n coloane de numere intregi. Dimensiunile matricei si elementele ei se gasesc in fisierul A.TXT astfel: pe prima linie se gasesc valorile m si n.. Pe urmatoarele m linii se gasesc elementele celor m linii ale matricei. Sa se verifice daca exista in matrice doua linii identice si in caz afirmativ sa se afiseze numerele de ordine ale celor doua linii, iar in caz contrar sa se afiseze mesajul 'NU'.
11. Scrieti un program care citește din fisierul text BAC.TXT un număr natural  $n$  ( $2 < n < 21$ ) si apoi n linii cu cate n numere întregi de cel mult 7 cifre ce formează un tablou bidimensional a. Sa se afișeze in fisierul text Rez.TXT diferența dintre suma elementelor de pe diagonala principala si suma elementelor de pe diagonala secundara a matricei a.

12. Scrieți programul care citește din fișierul text BAC.TXT un număr natural  $n$  ( $n < 100$ ) și un șir cu  $n$  numere întregi din intervalul  $[100; 999]$ ; programul construiește un șir de numere rezultat prin înlocuirea fiecărui număr din șirul citit cu numărul obținut prin interschimbarea cifrei unitatilor cu cifra sutelor. Numerele din noul șir se vor afișa în fișierul text Rez.TXT separate printr-un singur spațiu. De exemplu, pentru  $n=3$  și șirul 123, 904, 500, se afișează 321, 409, 5.
13. Doua tablouri unidimensionale  $a$  și  $b$ , cu elementele  $a_1, a_2, \dots, a_n$ , respectiv  $b_1, b_2, \dots, b_n$  sunt în relația  $a \leq b$  dacă:  $a_1 \leq b_1, a_2 \leq b_2, \dots, a_n \leq b_n$ . Scrieți program care citește din fișierul text BAC.TXT două tablouri unidimensionale  $a$  și  $b$  cu același număr de elemente de tip întreg și verifică dacă  $a \leq b$  sau  $b \leq a$  afișând în fișierul text Rez.TXT un mesaj adecvat.
14. Scrieți un program care afișează în fișierul text Rez.TXT toate numerele naturale formate din cifre identice, mai mari decât 10 și mai mici decât o valoare dată  $n$ , citită din fișierul text BAC.TXT,  $n \leq 2.000.000$ . De exemplu pentru  $n=195$ , se afișează: 11, 22, 33, 44, 55, 66, 77, 88, 99, 111.
15. Scrieți un program C/C++ care citește din fișierul text BAC.TXT trei numere naturale  $x, y$  și  $k$ , ( $1 < x < y < 2000000$ ,  $k < 1000$ ) și afișează în fișierul text Rez.TXT  $k$  numere prime din intervalul  $[x, y]$ . Dacă nu există  $k$  numere prime în intervalul  $[x, y]$  se vor afișa toate numerele prime găsite iar pe linia următoare se va afișa mesajul "s-au găsit mai puține numere prime:" urmat de numărul acestora. De exemplu, pentru  $x=3, y=12$  și  $k=5$  se vor afișa pe ecran:  
3 5 7 11 s-au găsit mai puține numere prime:4
16. Scrieți un program care citește din fișierul text BAC.TXT, cel mult 100 de numere naturale aflate pe o singură linie, formate din cel mult 9 cifre fiecare, separate prin spații și dintre acestea le afișează în fișierul text Rez.TXT doar pe acelea care au proprietatea de a fi palindrom. Dacă nu se găsesc numere palindrom, se va afișa pe ecran valoarea -1. Un număr are proprietatea de a fi palindrom dacă citit de la dreapta la stânga sau de la stânga la dreapta are aceeași valoare. De exemplu 1221 este palindrom, în timp ce 1210 nu este palindrom.  
  
Exemplu: dacă din fișierul BAC.TXT se citesc numerele: 7341, 8228, 660, 2, 80, 131, atunci pe ecran se vor afișa: 828, 2, 131.
17. Pentru orice număr natural nenul  $n$  definim  $n$  factorial (notat  $n!$ ) ca fiind produsul tuturor numerelor naturale nenule mai mici sau egale cu  $n$  ( $n! = 1 * 2 * 3 * \dots * n$ ). De exemplu:  $3! = 1 * 2 * 3 = 6$ ,  $5! = 1 * 2 * 3 * 4 * 5 = 120$ . Scrieți un program care determină numărul de cifre nule aflate pe ultimele poziții consecutive ale valorii obținute în urma evaluării lui  $n!$ ,  $n$  fiind un număr natural de cel mult 4 cifre citit din fișierul text BAC.TXT. De exemplu dacă  $n=10$ ,  $n!=3628800$  rezultatul va fi 2 deoarece 3628800 are două zerouri la sfârșit.
18. Scrieți un program care citește din fișierul text DATE.IN cel mult 100 de numere naturale nenule aflate pe o singură linie, formate din cel mult 4 cifre fiecare, separate prin spații și scrie în fișierul DATE.OUT numerele, în ordinea inversă față de cea în care au fost citite, pe o singură linie separate prin spații. De exemplu dacă din fișierul DATE.IN se citesc numerele 93 207 15 1982 3762, atunci conținutul fișierului DATE.OUT va fi 3762 1982 15 207 93.

19. Scrieți un program care citește de pe prima linie a fișierului text BAC.TXT trei numere naturale a, b, c formate din cel mult patru cifre fiecare, separate prin câte un spațiu și afișează pe ecran cel mai mare divizor comun al acestora. De exemplu, dacă din fișier se citesc numerele : 9 27 15, atunci se afișează 3.
20. Se citește de la tastatură un număr natural nenul n ( $n < 1000$ ). Scrieți un program care construiește fișierul text bac.txt care să conțină, pe prima linie, toți divizorii lui n în ordine strict descrescătoare. Divizorii vor fi separați prin spațiu. De exemplu, dacă  $n=10$ , atunci fișierul bac.txt va conține : 10 , 5 , 2 , 1
21. Se citește de la tastatură un număr natural nenul n care are cel mult 9 cifre. Să se afișeze în fișierul Date.out numărul k, natural, astfel încât produsul  $1*2*...*(k-1)*k$  să aibă o valoare cât mai apropiată de numărul n. De exemplu, dacă se citește numărul  $n=25$  fișierul Date.out are următorul conținut: 4. iar dacă se citește numărul  $n=119$  fișierul Date.out are următorul conținut: 5
22. Scrieți un program care citește un număr natural  $n > 1$  cu maximum 9 cifre din fișierul DATE.IN, și afișează în fișierul DATE.OUT valoarea celui mai mic divizor prim a lui n, precum și puterea la care acest divizor apare în descompunerea în factori primi a numărului n.
23. Să se scrie un program care să calculeze și să afișeze în fișierul DATE.OUT produsul cifrelor pare și suma cifrelor impare a unui număr natural n citit din fișierul text BAC.TXT
24. Să se verifice dacă două cuvinte citite din fișierul text BAC.TXT (cuvintele având maxim 50 de caractere fiecare) sunt rime, adică ultimele p caractere ale celor două cuvinte coincid, unde p ( $2 \leq p \leq 10$ ) este dat
25. Scrieți un program care construiește o matrice pătrată de ordin n formată din valorile 1 și 2 astfel încât elementele de pe diagonala principală și secundară să fie egale cu 1 iar restul elementelor cu 2. Valoarea lui n se citește din fișierul text BAC.TXT: n număr natural ( $2 < n < 23$ ), iar matricea se va afișa în fișierul text ies.txt, câte o linie a matricei pe fiecare rând cu spații între elementele fiecărei linii.
26. Scrieți un program care citește din fișierul text BAC.TXT un șir de cel mult 50 de caractere și construiește fișierul atestat.txt ce conține șirul și prefixele acestuia de lungime cel puțin 1, fiecare pe câte o linie, în ordinea descrescătoare a lungimii prefixelor.
27. Să se scrie un program care să numere cuvintele dintr-un text citit din fișierul "cuvinte.txt". Textul conține cuvintele separate numai printr-un spațiu, fără semne de punctuație, singurul semn de punctuație este '.' de la sfârșitul textului. Numărul de cuvinte se va afișa pe ecran.
28. Să se scrie un program care să numere cuvintele dintr-un text citit din fișierul "cuvinte.txt". Textul conține cuvintele separate numai printr-un spațiu, fără semne de punctuație, singurul semn de punctuație este '.' de la sfârșitul textului. Numărul de cuvinte se va afișa pe ecran.

29. Să se scrie un program care să numere cuvintele dintr-un text citit din fișierul "cuvinte.txt". Textul conține cuvintele separate numai printr-un spațiu, fără semne de punctuație, singurul semn de punctuație este '.' de la sfârșitul textului. Numărul de cuvinte se va afișa pe ecran. Exemplu: Fișierul "cuvinte.txt" conține textul: Ana are mere. se afișează: 3

30. Scrieți programul care citește din fișierul text atestat.txt o valoare naturală  $n$  ( $2 \leq n \leq 100$ ), construiește în memorie și apoi afișează în fișierul text ies.txt o matrice  $A$  cu  $n$  linii și  $n$  coloane, numerotate de la 1 la  $n$ , care conține numerele naturale, în ordine crescătoare, de la 1 la  $n^2$ , dispuse pe coloane, în ordine crescătoare. Astfel coloana 1 va conține numerele de la 1 la  $n$ , coloana 2 numerele de la  $n+1$  la  $2*n$ , coloana 3 de la  $2*n+1$  la  $3*n$  și așa mai departe, ca în exemplu. Matricea se va afișa în fișierul text ies.txt, câte o linie a matricei pe câte o linie, elementele fiecărei linii fiind separate între ele prin câte un spațiu.

Exemplu: pentru  $n = 4$  se va afișa matricea:

```
1 5 9 13
2 6 10 14
3 7 11 15
4 8 12 16
```

31. Se dă  $n$  număr natural. Să se afișeze în fișierul text ies.txt cel mai apropiat număr prim față de  $n$ . Programul va conține cel puțin un subprogram.

Exemplu:  $n=24$  se va afișa 23, pentru  $n=26$  se va afișa 29

32. Se considera un vector cu  $n$  ( $1 < n < 100$ ) componente numere naturale. Să se determine componentele vectorului cu cei mai mulți divizori și să se afișeze în fișierul text ies.txt. Se va folosi un subprogram pentru citirea vectorului și unul pentru calculul numărului de divizori ai unui număr.

33. Fișierul text 'date.txt' conține un text scris cu litere mari pe una sau mai multe linii. Scrieți un program care să determine acea literă care apare de cele mai multe ori. Dacă există mai multe astfel de litere, se vor afișa toate.

Exemplu: Dacă fișierul are conținutul "VARA ESTE BINE AICI" literele afișate sunt a, e și i.

34. Fiind date două numere naturale  $n$  și  $k$ , să se determine, recursiv, combinații de  $n$  luate câte  $k$ .

35. Dintre primele  $n$  numere naturale, să se afișeze acelea cu proprietatea: suma cifrelor lor este impară. Se va folosi o funcție care returnează suma cifrelor unui număr natural dat ca parametru.

36. Scrieți un program care generează toate numerele prime strict mai mici decât  $x$  ( $x$  număr natural). Valoarea variabilei  $x$  se citește de la tastatură. Numerele prime generate vor fi scrise în fișierul 'nr.txt', câte unul pe linie.

Exemplu: pentru  $n=10$  se vor scrie în fișier numerele 2, 3, 5 și 7.

37. Se citește de la tastatură o matrice  $A$  cu  $m$  linii și  $n$  coloane și elemente numere întregi. Să se determine linia (liniile) din matrice care conțin cele mai multe elemente nenule. Se va folosi

o functie care returneaza numarul elementelor nenule de pe o linie a carui indice i, se transmite ca parametru.

38. Pentru numerele complexe se defineste o structura cu doua componente : partea reala si partea imaginara.

a) Sa se scrie un program care calculeaza si afiseaza suma si produsul a doua numere complexe:  $z_1 = a_1 + ib_1$ ,  $z_2 = a_2 + ib_2$ .

b) Sa se scrie un program care determina si afiseaza maximul dintre modulele a n numere complexe  $z_1, z_2, z_3, \dots, z_n$ . Modulul unui numar complex  $z_k = a_k + ib_k$ ,  $|z_k| = \sqrt{a_k^2 + b_k^2}$ ,  $k=1, 2, \dots, n$ .

39. Fişierul atestat.in conţine două linii. Pe prima linie este scris un număr natural nenul n, ( $5 < n < 30$ ). Pe cea de-a doua linie a fişierului sunt scrise n numere naturale separate prin câte un spaţiu, formate fiecare din cel mult 4 cifre, reprezentând un şir de n numere naturale distincte.

Să se scrie un program în limbajul Pascal/C/C++, care:

- să afişeze pe ecran, în linie, toate numerele din şir, separate prin câte un spaţiu;
- să afişeze pe ecran, pe linii diferite, cel mai mic număr din şir şi poziţia acestuia;
- să scrie în fişierul atestat.out, pe o linie, separate prin câte un spaţiu, toate numerele *perfecte* din şirul dat (dacă nu există astfel de numere, se va se va afişa mesajul “NU EXISTĂ NUMERE PERFECTE”). Un număr este *perfect* dacă este egal cu suma divizorilor lui pozitivi, exceptându-l pe el însuşi, de exemplu:  $6 = 1 + 2 + 3$ .

Notă: Programul va conţine cel puţin un subprogram definit de utilizator.

Exemplu:

atestat.in						Date de ieşire:						
6						a)	28	11	81	496	6	100
28	11	81	496	6	100	b)	6					
							5					
							Fişierul atestat.out conţine:					
						c)	28	496	6			

40. Fişierul atestat.in conţine două linii. Pe prima linie este scris un număr natural nenul n, ( $5 < n < 30$ ). Pe cea de-a doua linie a fişierului sunt scrise n numere naturale separate prin câte un spaţiu, formate fiecare din cel mult 4 cifre, reprezentând un şir de n numere naturale distincte. Să se scrie un program în limbajul

Pascal/C/C++, care să afişeze pe ecran, în linie, toate numerele prime din şir, separate prin câte un spaţiu.

Notă: Programul va conţine cel puţin un subprogram definit de utilizator.

Să afişeze pe ecran, pe linii diferite, cel mai mic număr din şir şi poziţia acestuia;

**INSPECTOR ŞCOLAR GENERAL,  
PROF. GHERGHE VALERIA**

**INSPECTOR ŞCOLAR PENTRU INFORMATICĂ,  
PROF. DANIELA IOANA TĂTARU**