

**SUBIECTE PROPUSE PENTRU  
EXAMENUL DE ATESTAT PROFESIONAL LA INFORMATICĂ  
Programare Pascal/C/C++  
Matematică informatică, intensiv informatică**

1. Se citesc din fișierul "numere.in", de pe 2 linii consecutive, 2 numere mari scrise în baza 10, fiecare având cel mult 50 de cifre. Să se scrie un program care să calculeze suma lor, folosind șiruri în care se memorează cifrele numerelor. Rezultatul se va afișa în fișierul "numere.out".

Exemplu:

<b>numere.in</b> 1000000000099999 200001	<b>numere.out</b> 1000000000300000
--	---------------------------------------

2. Se citesc din fișierul "pol.in", de pe prima linie, gradele a două polinoame  $n, m$  ( $0 \leq n, m \leq 50$ ). De pe următoarele două linii se citesc două șiruri de numere reale care reprezintă coeficienții celor două polinoame, în ordinea crescătoare a gradelor. Să se scrie un program care să construiască în memorie și să afișeze în fișierul "pol.out" suma polinoamelor. Coeficienții sumei se vor afișa cu două zecimale separați printr-un singur spațiu.

Exemplu:

<b>pol.in</b> 2 3 7 -1 1 5 3 -4 1	<b>pol.out</b> 3 12 2 -3 1
--	----------------------------------

3. Se citesc din fișierul "pol.in", de pe prima linie, gradele a două polinoame  $n, m$  ( $0 \leq n, m \leq 50$ ). De pe următoarele două linii se citesc două șiruri de numere reale care reprezintă coeficienții celor două polinoame, în ordinea crescătoare a gradelor. Să se scrie un program care să construiască în memorie și să afișeze în fișierul "pol.out" produsul polinoamelor. Coeficienții produsului se vor afișa cu două zecimale separați printr-un singur spațiu .

Exemplu:

<b>pol.in</b> 3 1 1 5 -2 1 2 1	<b>pol.out</b> 4 2 11 1 0 1
---	-----------------------------------

4. Se citesc din fișierul "sort.in", de pe prima linie, 2 numere  $n$  și  $m$  ( $1 \leq n, m \leq 50$ ). De pe următoarele 2 linii se citesc elementele a două șiruri de lungime  $n$ , respectiv  $m$ , , numere întregi, de maxim 9 cifre, care sunt ordonate crescător. Să se scrie un program care să construiască în memorie un șir care conține rezultatul interclasării celor două șiruri. Șirul rezultat se va afișa în fișierul "sort.out".

Exemplu:

<b>sort.in</b> 5 4 1 3 4 7 8 1 2 3 6	<b>sort.out</b> 1 1 2 3 3 4 6 7 8
---	--------------------------------------

5. Se citesc din fișierul "mult.in", de pe prima linie, 2 numere  $n$  și  $m$  ( $1 \leq n, m \leq 50$ ). De pe următoarele două linii se citesc elementele a două șiruri de lungime  $n$ , respectiv  $m$ , numere întregi, de maxim 9 cifre. Elementele șirurilor sunt distincte două câte două. Scrieți un program, care să construiască în memorie un șir ce va conține intersecția celor două șiruri. Rezultatul se va afișa în fișierul "mult.out".

Exemplu:

<b>mult.in</b>	<b>mult.out</b>
5 4	1 3
1 3 4 7 8	
1 2 3 6	

6. Se citesc din fișierul "mult.in", de pe prima linie, 2 numere  $n$  și  $m$  ( $1 \leq n, m \leq 50$ ). De pe următoarele două linii se citesc elementele a două șiruri de lungime  $n$ , respectiv  $m$ , numere întregi, de maxim 9 cifre. Elementele șirurilor sunt distincte două câte două. Să se scrie un program care să construiască în memorie un șir ce va conține diferența dintre primul șir și cel de-al doilea șir. Rezultatul se va afișa în fișierul "mult.out".

Exemplu:

<b>mult.in</b>	<b>mult.out</b>
5 4	4 7 8
1 3 4 7 8	
1 2 3 6	

7. Se citesc din fișierul "mult.in", de pe prima linie, 2 numere  $n$  și  $m$  ( $1 \leq n, m \leq 50$ ). De pe următoarele două linii se citesc elementele a două șiruri de lungime  $n$ , respectiv  $m$ , numere întregi, de maxim 9 cifre. Elementele șirurilor sunt distincte două câte două. Să se scrie un program, care să construiască în memorie un șir ce va conține reuniunea celor două șiruri. Rezultatul se va afișa în fișierul "mult.out".

Exemplu:

<b>mult.in</b>	<b>mult.out</b>
5 4	1 2 3 4 6 7 8
1 3 4 7 8	
1 2 3 6	

8. Se citește din fișierul "mult.in", de pe prima linie, un număr  $n$  ( $1 \leq n \leq 50$ ). De pe următoarea linie se citesc elementele unui șir de lungime  $n$ , numere întregi, de maxim 9 cifre. Să se scrie un program care să verifice dacă șirul citit conține numai elemente distincte două câte două.

Exemplu:

<b>mult.in</b>	
5	da
1 3 4 7 8	

9. Se citește din fișierul “sort.in”, de pe prima linie, un număr  $n$  ( $1 \leq n \leq 50$ ). De pe următoarea linie se citesc elementele unui șir de lungime  $n$ , numere întregi, de maxim 9 cifre. Să se scrie un program care să realizeze sortarea crescătoare a elementelor șirului folosind metoda selecției. Rezultatul se va afișa în fișierul “sort.out”.

Exemplu:

<b>sort.in</b> 7 3 8 4 1 9 2 4	<b>sort.out</b> 1 2 3 4 4 8 9
--------------------------------------	----------------------------------

10. Se citește din fișierul “sort.in”, de pe prima linie, un număr  $n$  ( $1 \leq n \leq 50$ ). De pe următoarea linie se citesc elementele unui șir de lungime  $n$ , numere întregi, de maxim 9 cifre. Să se scrie un program care să realizeze sortarea crescătoare a elementelor șirului folosind metoda inserției. Rezultatul se va afișa în fișierul “sort.out”.

Exemplu:

<b>sort.in</b> 7 3 8 4 1 9 2 4	<b>sort.out</b> 1 2 3 4 4 8 9
--------------------------------------	----------------------------------

11. Se citește din fișierul “sort.in”, de pe prima linie, un număr  $n$  ( $1 \leq n \leq 50$ ). De pe următoarea linie se citesc elementele unui șir de lungime  $n$ , numere întregi, de maxim 4 cifre. Să se scrie un program care să realizeze sortarea crescătoare a elementelor șirului folosind metoda numărării. Se va ține cont de faptul că în șir poate să apară un element și de mai multe ori. Rezultatul se va afișa în fișierul “sort.out”.

Exemplu:

<b>sort.in</b> 9 3 8 -5 4 1 -7 9 2 4	<b>sort.out</b> -7 -5 1 2 3 4 4 8 9
--	--

12. Se citește din fișierul “cmmdc.in”, de pe prima linie, un număr natural  $n$  ( $1 \leq n \leq 50$ ). De pe următoarea linie se citesc  $n$  numere naturale de maxim 9 cifre.. Să se scrie un program care să calculeze cel mai mare divizor comun al celor  $n$  numere de pe linia a doua a fișierului. Rezultatul se va afișa în fișierul “cmmdc.out”.

Exemplu:

<b>cmmdc.in</b> 5 14 8 22 4 32	<b>cmmdc.out</b> 2
--------------------------------------	-----------------------

13. Se citește din fișierul “exista.in”, de pe prima linie, un număr  $n$  ( $1 \leq n \leq 50$ ). De pe următoarea linie se citesc  $n$  numere naturale de maxim 9 cifre reprezentând elementele unui șir. Să se scrie un program care să verifice dacă există în șir trei elemente alăturate care au același număr de divizori.

Exemplu:

<b>exista.in</b> 8 9 4 15 6 22 12 3 14	Da
--	----

14. Se citește din fișierul “progresie.in”, de pe prima linie, un număr  $n$  ( $1 \leq n \leq 50$ ). De pe următoarea linie se citesc elementele unui șir de lungime  $n$ , numere întregi, de maxim 9 cifre.. Să se scrie un program care să verifice dacă elementele șirului formează o progresie aritmetică sau o progresie geometrică. În caz afirmativ să se afișeze tipul progresiei și rația, iar în caz negativ să se scrie un mesaj corespunzător în fișierul ”progresie.out”.

Exemplu:

<b>progresie.in</b> 5 2 6 18 54 162	<b>progresie.out</b> elementele șirului formează o progresie geometrică cu rația 3
---	--

15. Se citește un text din fișierul ”text.in”. Textul are mai multe linii și mai multe cuvinte. Pe fiecare linie cuvintele sunt despărțite prin câte un singur spațiu sau prin semne de punctuație. Să se afișeze în fișierul ”text.out”, pe linii consecutive, toate literele care apar în text, împreună cu frecvențele lor de apariție.

Exemplu:

<b>text.in</b> Mama are un mar. Ce mare e marul!	<b>text.out</b> A – 6 C – 1 E – 4 M – 5 N – 1 R – 4 U - 2
--	--

16. Să se afișeze în fișierul “prietene.out” toate perechile de numere prime prietene mai mici decât un număr natural  $n$  citit de la tastatură ( $3 \leq n \leq 30000$ ). Două numere prime se numesc prietene dacă diferența lor în modul este mai mică sau egală cu 2.

Exemplu:

20	<b>prietene.out</b> (2 3), (3 5), (5 7), (11 13), (17 19)
----	--

17. Se citește un număr natural  $n$  de la tastatură ( $1 \leq n \leq 30000$ ). Să se verifice dacă  $n$  este număr puternic. Un număr este puternic dacă are mai mulți divizori decât orice număr mai mic decât el.

Exemplu:

6	DA
10	NU

18. Se citește de la tastatură un număr natural  $k$  ( $1 \leq k \leq 50$ ). Să se determine cel mai mic număr natural care are exact  $k$  divizori.

Exemplu:

5	16
---	----



19. Se citește din fișierul “mat.in”, de pe prima linie, un număr natural  $n$  ( $1 \leq n \leq 50$ ) și de pe următoarele  $n$  linii o matrice pătratică de numere întregi, de maxim 9 cifre. Să se verifice dacă matricea este pătrat magic (sumele elementelor pe linii, coloane și de pe cele două diagonale, este aceeași).

Exemplu

<b>mat.in</b> 3 4 3 8 9 5 1 2 7 6	da
---	----

20. Se citește din fișierul “mat.in”, de pe prima linie, un număr natural  $n$  ( $1 \leq n \leq 50$ ), și de pe următoarele  $n$  linii o matrice pătratică ce conține numere naturale de maxim 4 cifre. Să se numere elementele cu număr par de cifre aflate în triunghiul de sub diagonala principală a matricei și să se afișeze rezultatul pe ecran.

Exemplu

<b>mat.in</b> 4 1 4 12 3 13 3 22 5 14 22 2 34 222 11 3 31	4
--	---

21. Se citește din fișierul “mat.in”, de pe prima linie, un număr natural  $n$  ( $1 \leq n \leq 50$ ), și de pe următoarele  $n$  linii o matrice pătratică ce conține numere naturale de maxim 4 cifre. Să se calculeze suma elementelor prime aflate în triunghiul de deasupra diagonalei principale a matricei și să se afișeze rezultatul pe ecran.

Exemplu

<b>mat.in</b> 4 2 4 12 3 15 3 22 5 16 22 2 34 222 11 3 31	8
--	---

22. Se citește din fișierul “mat.in”, de pe prima linie, un număr natural  $n$  ( $1 \leq n \leq 50$ ), și de pe următoarele  $n$  linii o matrice pătratică ce conține numere naturale de maxim 4 cifre. Să se verifice dacă există elemente cu suma cifrelor pară în triunghiul de sub diagonala secundară a matricei și să se afișeze rezultatul pe ecran.

Exemplu

<b>mat.in</b> 4 7 11 5 8 9 5 1 12 3 4 31 15 3 2 23 14	Da
--	----

23. Se citește din fișierul “mat.in”, de pe prima linie, un număr natural  $n$ , ( $1 \leq n \leq 50$ ) și de pe următoarele  $n$  linii o matrice pătratică ce conține numere naturale de maxim 4 cifre. Să se verifice dacă toate elementele aflate în triunghiul de deasupra diagonalei secundare a matricei au aceeași paritate și să se afișeze rezultatul pe ecran.

Exemplu

<b>mat.in</b>	<b>Da</b>
4	
7 11 5 8	
9 5 1 12	
3 4 31 15	
3 2 23 14	

24. Se citesc din fișierul “mat1.in” de pe prima linie, două numere naturale  $n_1$  ( $1 \leq n_1 \leq 50$ ) și  $m_1$  ( $1 \leq m_1 \leq 50$ ) și de pe următoarele  $n_1$  linii o matrice cu  $n_1$  linii și  $m_1$  coloane. Analog, se citesc din fișierul “mat2.in”, de pe prima linie, două numere naturale  $n_2$  ( $1 \leq n_2 \leq 50$ ) și  $m_2$  ( $1 \leq m_2 \leq 50$ ) și de pe următoarele  $n_2$  linii o matrice cu  $n_2$  linii și  $m_2$  coloane. Elementele celor 2 matrice sunt numere reale. Să se scrie un program în care se calculează produsul celor două matrice. Dacă produsul se poate calcula atunci se va afișa matricea rezultată în fișierul “mat.out”, iar în caz contrar se va afișa un mesaj corespunzător.

Exemplu

<b>mat1.in</b>	<b>mat2.in</b>	<b>mat.out</b>
2 3	3 2	17 7
3 1 2	5 1	28 12
5 2 3	0 2	
	1 1	

25. Se citesc din fișierul “mat.in”, de pe prima linie, două numere naturale  $n$  și  $m$  ( $1 \leq n, m \leq 50$ ) și de pe următoarele  $n$  linii o matrice cu  $n$  linii și  $m$  coloane cu elemente numere întregi, de maxim 9 cifre. Să se scrie un program care afișează în fișierul “mat.out”, câte unul pe linie, numerele din matrice care sunt în același timp maxime pe linia și minime pe coloana pe care se află. Dacă nu există astfel de numere se va afișa un mesaj corespunzător.

Exemplu

<b>Mat.in</b>	<b>mat.out</b>
3 4	3
1 3 2 2	
2 7 8 1	
4 8 5 2	

26. Se citesc din fișierul “mat.in”, de pe prima linie, două numere naturale  $n$  și  $m$  ( $1 \leq n, m \leq 50$ ) și de pe următoarele  $n$  linii o matrice cu  $n$  linii și  $m$  coloane cu elemente numere naturale, de maxim 9 cifre. Să se scrie un program care afișează în fișierul “mat.out”, câte unul pe linie, numerele din matrice care au număr maxim de divizori

Exemplu

<b>Mat.in</b>	<b>mat.out</b>
3 4	14
9 3 14 10	10
8 5 6 21	8
11 3 15 17	6
	21
	15

27. Se citesc din fișierul “mat.in” , de pe prima linie, patru numere naturale  $n, m, l$  și  $c$  ( $1 \leq n, m \leq 50$  ,  $1 \leq l \leq n$ ,  $1 \leq c \leq m$ ) și de pe următoarele  $n$  linii o matrice cu  $n$  linii și  $m$  coloane, cu elemente numere întregi, de maxim 9 cifre. Să se scrie un program care să șteargă în memorie din matrice linia  $l$  și coloana  $c$  și afișează matricea rezultată în fișierul “mat.out”.

Exemplu

<b>Mat.in</b>	<b>mat.out</b>
3 4 2 3	1 2 4
1 2 3 4	9 10 12
5 6 7 8	
9 10 11 12	

28. Se citește un număr natural  $n$  ( $1 \leq n \leq 15$ ). Să se afișeze în fișierul “Pascal.out” triunghiul lui Pascal care are  $n + 1$  linii și  $n + 1$  coloane. Regulile după care se construiește triunghiul lui Pascal sunt:
- Pe coloana 0 și pe diagonala principală se pune 1.
  - Restul elementelor din triunghi se obțin adunând elementul aflat pe linia anterioară pe aceeași coloană cu el cu elementul aflat pe linia anterioară și pe coloana anterioară.

Exemplu

3	<b>Pascal.out</b>
	1
	1 1
	1 2 1
	1 3 3 1

29. Se citesc de la tastatură numitorii și numărătorii (numere întregi de maxim 9 cifre) a două fracții. Să se scrie un program care simplifică suma celor două fracții și afișează pe ecran numitorul și numărătorul fracției rezultate.

Exemplu

1 2 1 3	5 6
---------	-----

30. Se citesc de la tastatură două cuvinte de maxim 20 de litere mici. Să se scrie un program care verifică dacă unul este anagrama celuilalt. Un cuvânt este anagramă pentru altul dacă cele două cuvinte conțin aceleași litere de același număr de ori, dar nu neapărat în aceeași ordine.

Exemplu

mare rame	Da
mare mere	Nu

31. Se citește din fișierul “sir.in”, de pe prima linie, un număr natural  $n$  ( $1 \leq n \leq 50$ ). De pe următoarea linie se citesc  $n$  numere naturale de maxim 9 cifre. Să se scrie un program care să afișeze pe ecran câte din elementele șirului sunt termeni ai șirului lui Fibonacci. (Sirul lui Fibonacci 1,1,2,3,5,8,13,...).

Exemplu

<b>sir.in</b>	4
7	
10 2 17 8 1 9 1	

32. Se citesc din fișierul “numere.in” doua numere naturale de maxim 9 cifre. Se cere să se afișeze pe ecran mesajul “DA”, dacă cele două numere sunt termeni consecutivi ai șirului lui Fibonacci, respectiv mesajul “NU”, dacă cele două numere nu sunt termeni consecutivi ai șirului lui Fibonacci. (Sirul lui Fibonacci 1,1,2,3,5,8,13,...).

Exemplu

<b>numere.in</b> 8 5	DA
21 13	DA
21 10	NU

33. Se citește din fișierul “sir.in”, de pe prima linie, un număr natural  $n$  ( $1 \leq n \leq 50$ ). De pe următoarea linie se citesc  $n$  numere naturale de maxim 9 cifre. Să se afișeze toate perechile de elemente egal depărtate de mijloc, care au aceeași suma a cifrelor. Se trateaza si cazul "NU EXISTA".

Exemplu

<b>sir.in</b> 9 72123 34 45 1000 456 10 326 234 456	72123 456 1000 10
---	----------------------

34. Se citește din fișierul “text.in” un șir cu maxim 255 de caractere, format doar din litere mici și spații. Se consideră cuvânt orice secvență formată doar din litere adiacente. Cuvintele sunt separate prin exact un spațiu. Să se afișeze cuvintele palindrom din text. Dacă nu există cuvinte palindrom atunci se va afișa mesajul “NU EXISTĂ”.

Exemplu

<b>text.in</b> aerisirea este blocata de un cojoc	aerisirea cojoc
--	--------------------

35. Se citește din fișierul “text.in” un șir cu maxim 255 de caractere, format doar din litere mici și spații. Se consideră cuvânt orice secvență formată doar din litere adiacente. Cuvintele sunt separate prin exact un spațiu. Să se afișeze în fișierul “rime.out” toate perechile de cuvinte care rimează. Două cuvinte rimează dacă au ultimele două litere identice. Se trateaza si cazul "NU EXISTA".

Exemplu

<b>text.in</b> zic cei de la sate ca soarele puternic ne arde	<b>rime.out</b> zic puternic de arde
--	--

36. Se citește din fișierul “text.in” un șir de maxim 255 de caractere. Să se determine cea mai lungă secvență de cifre alăturate din șir. Secvența găsită se va afișa pe ecran.

Se trateaza si cazul "NU EXISTA".

Exemplu:

<b>text.in</b> A12abac12223defg895	12223
---------------------------------------	-------

37. Se citește din fișierul “text.in” un șir cu maxim 255 de caractere, format doar din litere (mici și mari) și spații. Se consideră cuvânt orice secvență formată doar din litere adiacente. Cuvintele sunt separate prin exact un spațiu. Să se afișeze cuvintele șirului dat, în ordine alfabetică.

Exemplu

<b>text.in</b> anul trecut aveam alte prioritati	alte anul aveam prioritati trecut
---	-----------------------------------

38. Se citește din fișierul “text.in” un șir cu maxim 255 de caractere, format doar din litere (mici și mari) și spații. Se consideră cuvânt orice secvență formată doar din litere adiacente. Cuvintele sunt separate prin exact un spațiu. Să se afișeze cuvintele șirului dat, în ordinea crescătoare a lungimilor lor, câte un cuvânt pe fiecare linie a ecranului.

Exemplu

<b>text.in</b> examen de atestat profesional la informatica	de la examen atestat profesional informatica
--	---

39. Fișierele X.txt si Y.txt conțin fiecare numele distincte a 7 persoane, câte un nume pe fiecare linie având cel mult 25 caractere. Știind că, în fiecare fișier numele sunt memorate în ordine alfabetică, scrieți un program care să citească din cele două fișiere numele și să afișeze pe ecran toate numele din cele două fișiere în ordine alfabetică, separate printr-un singur spațiu. Dacă un nume se află în ambele fișiere, atunci el se va afișa o singură dată.

Exemplu

<b>X.txt</b> Ana Dana Daniel Ene Mara Nae Paul	<b>Y.txt</b> Angi Cora Dora Horia Oana Paul Tibi	Ana Angi Cora Dana Daniel Dora Ene Horia Mara Nae Oana Paul Tibi
---	---	---

40. Se citesc două triplete de numere naturale (**d1,m1,y1**), respectiv (**d2,m2,y2**). Să se verifice dacă cele două triplete reprezintă două date calendaristice valide, în care **d1,d2** = ziua (day), **m1,m2** = luna (month), **y1,y2** = anul (year). Se consideră an bisect orice an care fie este divizibil cu 400, fie este divizibil cu 4 dar nu și cu 100.

Dacă cele două triplete reprezintă două date calendaristice valide, atunci în ipoteza că prima dată reprezintă ziua de naștere a unei persoane, iar a doua dintre ele reprezintă data curentă, să se determine, în ani împliniți, vârsta persoanei la data curentă.

Se garantează că prima dintre date este anterioară celei de-a doua.

Exemplu

29 2 2001 30 5 2013	data invalida
25 4 1995 20 5 2013	18 ani
25 6 1995 10 5 2013	17 ani