

Atestat 2013. Subiecte programare

1. Se citește de la tastatură n numere naturale, cu cel mult 9 cifre fiecare. Scrieți un program care pentru o cifră k citită de la tastatură, afișează pe ecran câte numere prime în scrierea cărora apare cifra k , se găsesc în șirul dat.

Exemplu: pentru $n=4$, cifra $k=2$ și valorile citite **23, 603, 122, 27** se obține numărul $nr = 1$, care corespunde valorii **23**.

2. Se citește de la tastatură un număr natural n cu cel mult nouă cifre. Scrieți un program care verifică dacă există un număr natural k cu proprietatea că $n=1*2*3*...*k$. Dacă există un astfel de număr, programul va afișa mesajul **“Da”** și valoarea k , altfel va afișa mesajul **“Nu”**. **(intensiv: valoarea se determină utilizând o metodă eficientă din punct de vedere al timpului de execuție)**

3. În fișierul **valori.txt** se găsesc pe prima linie maximum 1000 de numere întregi despărțite printr-un spațiu. Scrieți un program care să determine valoarea maximă și valoarea minimă, împreună cu numărul de apariții al maximumului, respectiv al minimumului. **(intensiv: folosiți o metodă eficientă din punct de vedere al memoriei și al numărului de operații efectuate și afișați pe ecran valorile obținute).**

Exemplu: conținutul fișierului **valori.txt** este : 24 3 89 24 1 3 1 1 89, atunci se vor afișa rezultatele:

min = 1 nr_aparitii = 3

max = 89 nr_aparitii = 2

4. Fișierul **unu.in** conține pe primul rând valoarea n ($2 \leq n \leq 1.000.000$), iar pe a doua linie se găsesc n numere întregi formate din cel mult 9 cifre și despărțite prin spațiu. Știind că fișierul conține cel puțin două numere distincte printre cele n de pe a doua linie, scrieți un program care să scrie în fișierul **unu.out** pe o singură linie și separate printr-un spațiu, în ordine descrescătoare, cele mai mari două valori distincte din fișierul de intrare. **(intensiv : alegeți o metodă eficientă de rezolvare atât ca timp de executare, cât și ca gestionare a memoriei)**

Exemplu :

unu.in	unu.out
10	
-8 14 8 14 15 9 -7 1 4 10	15 14

5. Fișierul text **numere.in** conține pe prima linie un număr natural n ($0 \leq n \leq 5000$), iar pe a doua linie n numere naturale de cel mult 9 cifre fiecare, separate prin câte un spațiu. Să se scrie un program care citește n , apoi cele n numere naturale din fișierul **numere.in** și scrie în fișierul **numere.out**, pe câte o

linie fiecare, numerele de pe a doua linie a fișierului **numere.out** care sunt palindroame cu exact patru cifre (un număr este palindrom dacă este egal cu oglinditul său).

De exemplu dacă fișierul **numere.in** are următorul conținut :

5

1441 5 14 2552 78

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

1441

2552

6. Se citește de la tastatură **n**, și apoi **n** perechi de câte două numere întregi **a** și **b** cu $(a < 0 < b)$ reprezentând capetele intervalului **[a, b]**. Afișați pe ecran, dacă există, capetele intervalului de intersecție al celor **n** intervale citite. În cazul în care **nu există** intervalul de intersecție se afișează **0**.

De exemplu, pentru **n=3** și perechile **[-1,20], [-4,9], [-7, 12]**, obținem soluția **x= -1 și y= 9**.

7. Se citesc de la tastatură **n** numere naturale, cu cel mult 9 cifre fiecare. Scrieți un program care afișează cifra care apare de cele mai multe ori în numerele citite. Dacă există mai multe astfel de cifre se vor afișa **toate**.

Exemplu: Pentru **n=4** și valorile 23, 431, 154, 452 se afișează 3 pentru că cifra 4 apare de 3 ori în numerele citite.

8. Se citește de la tastatură un număr natural **n** cu maxim cinci cifre. Afișați cel mai apropiat număr față de **n** care are aceeași sumă a cifrelor.

9. Se citesc **n** numere naturale. Să se ordoneze numerele crescător după numărul de divizori. (**intensiv: numărul de divizori se va determina utilizând o metodă eficientă din punct de vedere al timpului de execuție**)

Exemplu: **n=4**, numerele 12,4,6,13 ⇔ 13,4,6,12 .

10. Se citește **n** număr natural ($n < 32000$) și apoi se introduc pe rând **n** numere întregi. Să se afișeze numerele prime și să se calculeze media lor aritmetică.

11. Se citește de la tastatură un număr natural **n** cu cel mult 9 cifre. Să se afișeze cel mai mare și cel mai mic număr care se poate forma cu cifrele distincte ale numărului.

12. Se citește din fișierul **numar.txt** un număr natural **n**. Să se verifice dacă este putere a lui 2 și să se calculeze frecvența de apariție a fiecărei cifre în scrierea în baza 10 a numărului.

13. Scrieți un program care afișează pe ecran toate numerele naturale cu maxim 9 cifre care sunt rotunde (numerele care convertite în binar au același număr de 0 respectiv de 1).

Exemplu: 12 ⇒ 1100 este rotund; 13 ⇒ 1101 nu este rotund)

14. Fișierul text **numere.in** conține pe prima linie un număr natural **n** ($0 < n < 1000$), iar pe a doua linie **n** numere naturale cu cel mult 9 cifre fiecare, despărțite prin câte un spațiu. Scrieți un program C/C++ care citește toate numerele din fișier și afișează pe ecran, separate prin câte un spațiu, numerele de pe a doua linie a fișierului, care sunt formate din cifre identice.

Exemplu: dacă fișierul **numere.in** are conținutul:

9

55 107 3 101 99 7 208 2222 80

numerele ce se vor afișa sunt:

55 3 99 7 2222

15. Fișierul **atestat.txt** conține pe prima linie un număr natural **n** ($n \leq 100$), iar pe a doua linie, separate prin câte un spațiu, **n** numere naturale nenule, cu cel mult 4 cifre fiecare. Scrieți programul C/C++ care afișează pe ecran cifra de control a fiecărui număr ($c(4568)=c(4+5+6+8)=c(23)=5$).

Exemplu: dacă fișierul are conținutul alăturat,

7

120 346 68 9 32 0 112

atunci se afișează 3 4 5 9 5 0 4

16. Fișierul **numere.in** conține pe prima linie mai multe numere naturale în ordine crescătoare dintre care cel puțin o putere a lui 3. Scrieți un algoritm **eficient** care afișează pe ecran numerele din fișier ce sunt puteri ale lui 3.

17. Dat **n** natural, sa se descompuna ca suma de puteri distincte ale lui 2.

Exemplu : $25=2^4+2^3+2^0$.

18. Să se scrie un program care să verifice dacă suma cifrelor de pe pozițiile impare și suma cifrelor de pe pozițiile pare ale unui număr natural **n** citit de la tastatură sunt egale. Pozițiile se numără de la dreapta la stânga începând cu poziția 1. Se afișează sumele și un mesaj corespunzător pe ecran.

Exemplu: se citește: $n=12345$ și

se afișează: $sp=6$ și $simp=9$ Nu coincid!

19. Să se afișeze primele **n** perechi de numere prime gemene, unde **n** este un număr natural nenul citit de la tastatură.

Exemplu: se citește: $n=4$

se afișează: (3 ; 5) ; (5 ; 7) ; (11 ; 13) ; (17 ; 19)

20. Se citesc **n** numere naturale nenule. Să se precizeze cu câte zerouri se va termina produsul numerelor fără a calcula efectiv produsul lor.

Exemplu: se citește: $n=3$ și numerele 25, 12, 9

se afișează: 2 de zero

21. Se citește un număr natural **n**. Să se afișeze toate numerele prime mai mici decât **n** astfel încât suma cifrelor a fiecărui număr să fie mai mică decât un alt număr citit **m**.

Exemplu: se citește: $n=30$ și $m=7$

se afișează: 2, 3, 5, 11, 13, 23

22. Fișierul **matrice.txt** conține pe primul rând două valori naturale **m** și **n** ($1 \leq n \leq 100$, $1 \leq m \leq 100$), reprezentând numărul de linii și respectiv de coloane ale unei matrice **a**, iar pe următoarele **m** linii câte **n** valori întregi cu maximum 4 cifre fiecare, separate prin câte un spațiu, reprezentând elementele matricei **a**. Afișați pe ecran un șir de $2 \cdot (n+m) - 4$ numere ordonate crescător, șir format din elementele aflate pe chenarul exterior al matricei **a**. Chenarul exterior este format din prima și ultima linie, respectiv prima și ultima coloană din matrice.

Exemplu: dacă matricea dată este :

3 4

1 2 3 4

5 6 7 8

9 1 2 3

se va afișa șirul : 1 1 2 2 3 3 4 5 8 9.

23. Se citește de la tastatură o valoare naturală m ($2 \leq m \leq 100$). Scrieți programul C/C++/Pascal care construiește în memorie și apoi afișează pe ecran matricea a cu m linii și m coloane de numere întregi, construită după următoarea regulă : elementul de pe linia i și coloana j a matricii ($1 \leq i, j \leq m$) este :

- 1 dacă $i*i+j*j$ este pătrat perfect
- 2 dacă $i*i+j*j$ nu este pătrat perfect, dar este număr prim mai mare decât 2
- 3 în rest.

Exemplu: pentru $m = 2$, se va afișa matricea :

```
3 2
2 3
```

24. Se citește un vector v cu n componente numere naturale. Să se afișeze cel mai mare număr care se poate forma cu cifra maximă a fiecărui număr din vector.

Exemplu: se citește: $n=5$ și $v=(29, 171, 0, 2222, 100)$

se afișează: 97210

25. Se consideră o matrice oarecare de dimensiune $m \times n$ cu elemente numere naturale. Să se afișeze numerele cu număr impar de divizori din matrice și să se indice numărul lor.

$$A = \begin{pmatrix} 4 & 11 & 8 \\ 12 & 7 & 25 \end{pmatrix}$$

Exemplu: se citește: $m=2$ și $n=3$ și matricea

se afișează: 4, 25,

2 numere

26. Fie un vector v cu n elemente numere întregi citite de la tastatură. Fără a folosi alt vector auxiliar, să se mute la sfârșitul lui v elementele sale nule, păstrând ordinea celorlalte elemente.

Exemplu: se citește: $n=7$ și vectorul $v=(2,3,0,9,0,0,8)$

se afișează: $v=(2,3,9,8,0,0,0)$

27. Fie a o matrice pătratică de dimensiune n . Folosind o funcție de ordonare a unui vector de dimensiune k , afișați matricea modificată prin ordonarea crescătoare a liniilor impare.

28. Fisierul **matrice.in** conține pe prima linie un număr natural n ($0 \leq n \leq 5000$), iar pe următoarele $2n$ linii câte n numere naturale de cel mult 5 cifre fiecare, separate prin câte un spațiu reprezentând elementele a două matrice a și b cu câte $n \times n$ elemente fiecare. Verificați dacă b este obținută din a prin rotirea spre stânga cu 270 de grade, afișând un mesaj afirmativ sau „nu”.

Exemplu:

Dacă fișierul **matrice.in** are următorul conținut :

```
3
1 2 3
3 4 5
5 6 7
```

```
5 3 1
6 4 2
7 5 3
```

atunci se va afișa “DA”

29. Dată o matrice cu elemente numere întregi de dimensiune $n \times m$ să se determine punctele-șă din matrice, adică elementele care sunt minime pe linia lor și maxime pe coloana lor. În cazul în care nu există se va afișa mesajul “NU”.

Exemplu=3;

```
4 3 6
```

```
12 1 4    => numarul 3 este șă.
```

```
5 0 7
```

30. Fie **n** și **m** două numere naturale citite de la tastatură. Să se formeze un tablou bidimensional cu **n** linii și **m** coloane cu primele **n**×**m** numere naturale **imperfecte**. Un număr natural **n** este perfect dacă suma **S** a divizorilor săi mai mici decât **n** este egală de numărul **n** (exceptând numărul însuși).

Exemplu: $n=2$, $m=3$ atunci matricea este

1 2 3

4 5 7

! 6 este număr perfect – $6=1+2+3$

31. Scrieți un program C/C++ care citește de la tastatură un număr natural **n** ($2 < n < 20$), construiește în memorie și afișează pe ecran o matrice cu **n** linii și **n** coloane, numerotate de la 1 la **n**. Fiecare element din matrice aflat pe o linie impară va fi egal cu numărul liniei pe care se află iar elementele de pe linii pare să fie completate cu valorile de la **n** la 1. Elementele matricei vor fi afișate pe ecran, câte o linie a matricei pe câte o linie a ecranului cu câte un spațiu între elementele fiecărei linii.

Exemplu: pentru $n=5$ se va afișa matricea de mai jos

1 1 1 1 1

5 4 3 2 1

3 3 3 3 3

5 4 3 2 1

5 5 5 5 5

32. Se dau numărul de **n** linii și **m** coloane a unei matrici. Să se construiască și să se afișeze matricea în fișierul matrice.out știind că elementul din poziția *i,j* este egal cu minimul dintre *i* și *j*. Afișarea se realizează prin intermediul unui vector construit prin parcurgerea matricei sub formă de L.

33. Fișierul text **NUMERE.IN** conține pe prima linie un număr natural nenul **n** ($1 \leq n \leq 100$) și pe următoarea linie **n** numere reale pozitive separate prin câte un spațiu. Scrieți un program C/C++ care citește din fișierul **NUMERE.IN** numărul natural **n**, și determină, utilizând un algoritm eficient din punct de vedere al timpului de executare și al memoriei utilizate, poziția pe care se află primul și ultimul număr din cele **n** numere reale care sunt egale cu valoarea **minimă** dintre ele.

Exemplu: Dacă fișierul **NUMERE.IN** are conținutul:

6

2.8 2.3 5.7 5.7 2.3 6.3

atunci se afișează **2 și 5** (valoarea minimă este 2.3 și numerele de pe poziția 2 și 5 sunt egale cu 2.3)

34. Scrieți un program C/C++ care citește de la tastatură un număr natural **n** ($1 \leq n \leq 20$), elementele unei matrici cu **n** linii și **n** coloane, numere întregi din intervalul $[-100, 100]$ și afișează pe ecran **m1** și **m2**, unde **m1** este media aritmetică a elementelor strict pozitive ale matricei, situate deasupra diagonalei principale, iar **m2** este media aritmetică a elementelor strict pozitive ale matricei, situate sub diagonala principală, ca în exemplu. Cele două medii se consideră egale cu 0 dacă nu există valori strict pozitive în zonele corespunzătoare.

Exemplu: pentru $n=4$ și matricea alăturată se afișează

$m1=2.75$, calculată din elementele aflate deasupra diagonalei

principale, și $m2=2.5$, calculată din elementele aflate sub diagonala principală.

-1 2 -4 5

0 6 3 1

2 4 2 0

3 -5 1 -3

35. Scrieți un program C/C++ care citește de la tastatură numerele întregi **m** și **n** ($1 \leq m \leq 50$, $1 \leq n \leq 50$) și elementele unui tablou bidimensional cu **m** linii și **n** coloane, numere întregi distincte de cel mult 4 cifre fiecare. Programul va afișa pe prima linie a ecranului numărul de elemente prime de pe fiecare coloana a matricii, separate prin câte un spațiu. Definiți o funcție care verifică dacă un număr întreg este prim și folosiți apeluri ale ei pentru rezolvarea cerinței.

Exemplu: pentru $m=3$ și $n=4$ și tabloul de mai jos

2 7 1 4

14 6 12 3

9 22 8 5

Pe ecran se va afișa:

3 1 1

36. Fie x un vector de numere întregi de lungime n , să se construiască un vector y , astfel încât $y[i]=$ numărul de apariții ale lui $x[i]$ în vectorul x .

Exemplu: Pentru $x=(1,5,2,1,5,7,2,1,5)$ se obține $y=(3,3,2,3,3,1,2,3,3)$.

37. Scrieți un program care citește de la tastatură două numere naturale nenule m și n și care construiește în memorie și apoi afișează o matrice A cu m linii și n coloane cu proprietatea că fiecare element a_{ij} memorează cea mai mică dintre valorile indicilor i și j . Matricea se va afișa în fișierul **matrice.txt**, câte o linie a matricei pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spațiu.

Exemplu: Pentru $m=4$ și $n=5$ fișierul va conține matricea alăturată

1 1 1 1 1

1 2 2 2 2

1 2 3 3 3

1 2 3 4 4

38. Se citesc de la tastatură doi vectori cu câte n elemente. Afișați ultima cifră a sumei $a_1 \cdot b_1 + a_2 \cdot b_2 + \dots + a_n \cdot b_n$ fără a calcula efectiv suma.

39. Scrieți un program care construiește în memorie o matrice pătratică de ordin n cu elemente numere naturale astfel încât pe diagonala principală să existe numai elemente egale cu 1, elementele de pe cele două « semidiagonale » paralele cu diagonala principală și alăturate diagonalei principale să fie toate egale cu 2, elementele de pe următoarele două « semidiagonale » să fie egale cu 3, etc.

Valoarea lui n se citește de la tastatură, iar matricea se va afișa pe ecran.

Pentru $n=4$ se va afișa tabloul :

1 2 3 4

2 1 2 3

3 2 1 2

4 3 2 1

39. Fie a un vector cu n componente reale ($n \geq 30$). Să se obțină din a vectorul b care să îndeplinească condițiile :

- $b[i] \geq b[i+1]$ pentru $i=1,2,\dots,k-1$

- $b[i] \leq b[i+1]$ pentru $i=k,k+1, \dots, n-1$

- $b[k-i] \geq b[k+i]$ pentru $i=1,2, \dots, k-1$, unde $k=n/2$ când n este par și $k=(n+1)/2$ când n este impar.

40. Se citește de la tastatură o matrice a cu m linii și n coloane, ale cărei elemente sunt cifre de la 0 la 9 ($0 \leq m, n \leq 5$). Să se afișeze suma numerelor care se pot forma cu cifrele fiecărei linii (numărul aferent fiecărei linii se va obține prin citirea cifrelor de pe linia respectivă de la stânga la dreapta, ignorându-se zerourile de la începutul liniei).

Exemplu : Pentru matricea :

0 3 8 2 5

0 0 5 1 4

0 7 7 9 3 se va afișa valoarea 12132 reprezentând suma $3825+514+7793$.

41. Scrieți un program care construiește în memorie o matrice t pătratică de ordin n cu elemente numere naturale astfel încât pe ultima coloană și pe ultima linie să se afle numai elemente egale cu 1, iar oricare

alt element al matricei să fie egal cu suma dintre elementul aflat imediat sub el și elementul aflat imediat în dreapta lui.

Exemplu : Pentru $n = 4$ se va afișa tabloul următor :

```
20 10 4 1
10 6 3 1
4 3 2 1
1 1 1 1
```

42. Scrieți programul C, C++ sau Pascal, care citește de la tastatură un șir s de cel mult 30 de caractere și un caracter c . Programul determină triplarea fiecărei apariții a caracterului c în s și scrie noul șir obținut în fișierul text **final.out**. (**neintensiv: șirul poate fi scris în fișier fără a fi construit efectiv în memorie**)

De exemplu, dacă se citește șirul : **ciocarlie** și caracterul c atunci fișierul va conține șirul : **ccciocccarlie**.

43. Fișierul **cuv_a.txt** conține mai multe cuvinte ordonate alfabetic. Pentru un cuvânt c dat de la tastatură generați fișierul **cuv_b.txt** conține cuvintele din primul fișier și cuvântul c , toate în ordine alfabetică.

44. Definiți structura fracție care memorează în câmpul a al structurii numărătorul, iar în câmpul b numitorul fracției. Citiți numitorul și numărătorul a n fracții. Pentru fiecare fracție să se verifice dacă este ireductibilă, în caz afirmativ să se afișeze, iar în caz contrar să se afișeze numitorul și numărătorul după simplificare.

45. Se consideră un text cu maximum 255 de caractere în care cuvintele sunt separate prin unul sau mai multe spații. Primul caracter din textul citit este o literă, iar cuvintele sunt formate numai din litere mici ale alfabetului englez. Scrieți un program C/C++ care citește de la tastatură textul și îl transformă înlocuind prima literă a fiecărui cuvânt cu litera mare corespunzătoare, restul caracterelor rămânând nemodificate. Textul astfel transformat va fi afișat pe ecran.

Exemplu: dacă de la tastatură se introduce textul: clasă elev școală se va afișa pe ecran: Clasă Elev Școală

46. Se dă o înregistrare cu numele cerc, în care sunt stocate abscisa și ordonata centrului cercului și raza acestuia. Pentru un n număr natural < 100 , citiți datele caracteristice a n cercuri, sortați structurile descrescător în funcție de diametru și afișați coordonatele centrelor cercurilor în ordinea obținută după ordonare.

47. Scrieți programul C/C++ care citește de la tastatură un număr de n șiruri de cel mult 40 de caractere, formate doar din litere mici ale alfabetului englez. Să se verifice dacă șirul de pe poziția i se termina cu același caracter cu care începe șirul de pe poziția $i+1$, pentru i de la 1 la $n-1$.

Se va afișa pe ecran mesajul **Sirurile citite verifica proprietatea** sau **Sirurile citite nu verifica proprietatea**

Exemplu: dacă se citește șirul $n=4$ și sirurile

Proba a acoperit tot!

atunci pe ecran se va afișa: Sirurile citite verifica proprietatea

48. Fișierul text **linii.txt** este alcătuit din mai multe linii de lungime variabilă, pe fiecare linie găsindu-se câte un cuvânt. Scrieți un program care afișează linia (liniile) de lungime maximă.

Exemplu: Pentru fișierul „linii.txt”:

Se va afișa:

informatica

informatica

isoscel

programator

triunghi

trapez

programator

caiet

49. Fie s_1 și s_2 două șiruri de caractere. Verificați dacă s_2 este o codificare „shift” a lui s_1 (o codificare „shift” se obține din s_1 prin adăgarea la codul ASCII al fiecărui caracter din s_1 a unei constante k) afișând în caz afirmativ valoarea lui k sau mesajul „NU” în caz contrar.

Ex: Pentru șirurile $s_1 = \text{acdrtu}$ și $s_2 = \text{ceftvx}$ se afișează $k=2$.

50. Scrieți un program care citește de la tastatură un șir de caractere ce conține litere mici și spații. Afișați cuvântul cu cele mai multe vocale (dacă există mai multe se va afișa unul singur). Există cel puțin un cuvânt ce conține vocale.

51. Scrieți programul C/C++ care citește de la tastatură un cuvânt de cel mult 15 litere mici ale alfabetului englez și care afișează pe ecran, pe linii distincte, cuvintele obținute prin ștergerea succesivă a vocalelor din cuvântul citit, de la stânga la dreapta, ca în exemplu de mai jos:

Exemplu: Dacă se citește cuvântul **examen** se afișează:

xamen

xmen

xmn