

Probleme propuse pentru examenul de atestat – clasa a 12-a

Profilul : matematică informatică neintensiv

1. Se citesc de la tastatura **n** numere naturale, cu cel mult 9 cifre fiecare. Scrieti un program care pentru o cifra **k** citita de la tastatura, afiseaza pe ecran cate numere prime in scrierea carora apare cifra **k**, se gasesc in sirul dat.

Exemplu.: pentru **n=4** , cifra **k=2** si valorile citite **23, 603, 122, 27** se obtine numarul **nr = 1**, care corespunde valorii **23** .

2. Se citeste de la tastatura un numar natural **n** cu cel mult noua cifre si care determina daca exista un numar natural **k** cu proprietatea ca **n=1*2*3*...*k**. Daca exista un astfel de numar, programul va afisa mesajul “**Da**” , altfel va afisa mesajul “**Nu**”.

3. Se citeste de la tastatura un numar natural **n**, **0<n<1000000**. Sa se afiseze pe ecran, daca exista, un numar natural care este strict mai mare decat **n** si care are exact aceleasi cifre ca si **n**. Daca se gaseste un astfel de numar afisati mesajul **Exista**, altfel se va afisa mesajul **Nu exista**.

De exemplu, pentru **n=165**, exista numarul **561** care satisface conditia din enunt, deci se va afisa mesajul **Exista**. Pentru **n=14589** , nu exista nici un numar care sa satisfaca conditia din enunt, deci se va afisa mesajul **Nu exista**.

5. Se considera numarul natural **a**, **a>10000**. Folosind descompunerea in factori primi a lui **a**, decideti daca acesta poate fi exprimat ca produs de doua numere prime si dati un mesaj corespunzator.

De exemplu, pentru **n=15**, exista numerele **3 si 5** prime, care satisfac conditia din enunt, deci se va afisa mesajul **Da**, iar pentru **n=16**, se va afisa mesajul **Nu**.

6. În fişierul **valori.txt** pe aceeaşi linie, despărţite printr-un spaţiu, se găsesc maximum 1000 de numere întregi. Scrieţi un program care să determine valoarea maximă şi valoarea minimă, împreună cu numărul de apariţii al maximumului, respectiv al minimumului. Folosiţi o metodă eficientă din punct de vedere al memoriei şi al numărului de operaţii efectuate şi afişaţi pe ecran valorile obţinute.

Exemplu :dacă conţinutul fişierului **valori.txt** este : 24 3 89 24 1 3 1 1 89, atunci se vor afişa rezultatele:

min = 1 nr_aparitii = 3

max = 89 nr_aparitii = 2

7. 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. 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

8. 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**.

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

9. Fișierele **unu.txt** și **doi.txt** conțin fiecare dintre ele, numere întregi distincte ordonate crescător. Creați fișierul **trei.txt** și apoi scrieți în el, câte una pe linie, valorile distincte din cele două fișiere, în ordine crescătoare.

De exemplu, dacă fișierele conțin :

unu.txt	doi.txt
2 4 7 8 42 100 145 200	3 7 11 40 290

Atunci **trei.txt** va conține valorle :

2 3 4 7 7 8 11 40 42 145 200 290

10. Se dă șirul recurent definit prin relațiile : $x(1)=1$; $x(2)=2$; $x(n)=2*x(n-1) - x(n-2)$, pentru $n>2$. Cunoscând valoarea n , număr natural citit de la tastatură cu $0<n<100.000$, scrieți programul care afișează în ordine crescătoare toate valorile de 2 cifre din șirul recurent.

11. Scrieți un program care citește de la tastatură un șir de caractere format din cel mult 50 de litere din alfabetul englez și care afișează mesajul “**DA**” dacă șirul este format din număr egal de vocale și consoane, respective mesajul “**NU**” în caz contrar.

De exemplu, dacă fișierul conține șirul: aaabbbccce , atunci pe ecran va fi afișat mesajul “**DA**”.

12. Scrieți un program care citește de la tastatură două șiruri de caractere, fiecare șir fiind format din cel mult 100 de litere mici din alfabetul englez, și care afișează mesajul “**DA**” dacă toate literele

din primul șir se găsesc în cel de-al doilea șir, nu neapărat în aceeași ordine și de același număr de ori, sau mesajul "NU" în caz contrar.

De exemplu, dacă primul șir este "exemplu", iar al doilea șir este "pzyeaxqeemtluss", programul trebuie să afișeze mesajul "DA" deoarece literele primului șir apar în cel de-al doilea șir.

13. În fișierul **numere.txt**, se află memorate, pe prima linie, două numere naturale, n și m despărțite de un spațiu ($1 \leq n \leq 10000$ și $m \leq n$), iar pe următoarea linie a fișierului, n numere reale despărțite prin câte un spațiu. Scrieți programul C/C++/Pascal, care citește din fișier datele existente și afișează pe cea de-a treia linie a fișierului, suma obținută din cele mai mari m elemente negative citite din fișier. În cazul în care fișierul nu conține cel puțin m numere negative, se va afișa valoarea 0.

De exemplu, dacă fișierul conține șirul de valori :

10 2

-8 5 1 -3 5 23 7 -81 46 -120

atunci pe ecran vor fi afișate valorile -3 -8.

14. Scrieți programul C/C++/Pascal care citește de la tastatură un număr natural n ($n \leq 100$) și un șir cu n numere întregi din intervalul $[100, 999]$; programul construiește în mod eficient din punctul de vedere al spațiului de memorie folosit, un șir de numere rezultat prin înlocuirea fiecărui număr din șirul citit cu numărul obținut prin interschimbarea cifrei unităților cu cifra sutelor. Numerele din noul șir se vor afișa pe ecran separate printr-un singur spațiu.

De exemplu, pentru $n=3$ și șirul **123 904 500**, se afișează : **321 409 5**.

15. Scrieți programul C/C++/Pascal care citește două numere naturale a și b ($a \leq b$) cu cel mult 4 cifre fiecare și tipărește toate numerele prime de 3 cifre din intervalul $[a, b]$.

De exemplu, pentru $a=20$ $b=45$, se afișează : 23 29 31 37 41 .

16. 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.

De 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.

17. 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.

De exemplu pentru $m = 2$, se va afișa matricea :

3 2

2 3

18. 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 inversul său).

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

5

1441 5 14 2552 78

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

1441

2552

19. Se citeste de la tastatura n , si apoi n perechi de cate doua numere intregi a si b cu ($a < 0 < b$) numere ce reprezinta capetele unor intervale inchise. Afisati pe ecran , doua numere x si y ce reprezinta capetele intervalului **intersectie** al celor n intervale date.

De exemplu, pentru $n=3$ si perechile $[-1,20]$, $[-4,9]$, $[-7, 12]$, obtinem solutia $x= -1$ si $y= 9$.

20. Se citesc de la tastatura doua numere naturale n si p ($1 < n < 1000$, $1 < p < 10$). Sa se afiseze pe ecran, cu spatii intre ele, acele numere naturale mai mici sau egale cu n , care au toate cifrele mai mici sau egale cu p .

Exemplu : fiind dat $n=20$ si $p=5$ obtinem 1, 2, 3, 4, 5, 10, 11, 12, 13, 14, 15.

21. Fie un sir de n numere intregi si un interval [a,b]. Scrieti un program care sa precizeze care este numarul de valori din sir care **nu** se gasesc in intervalul dat.

De exemplu, pentru **n=10** si **[10 , 420]** si sirul de valori **1, 4, 7, 23, 90, 200, 440, 500, 501, 502**, programul va afisa valoarea **7**.

22. Fişierul text **cuvinte.in** conţine mai multe linii nevide de text, fiecare linie de cel mult 255 de caractere. Orice linie este compusă din unul sau mai multe cuvinte separate prin câte un spaţiu. Să se scrie un program care citeşte de la tastatură un număr natural L şi scrie în fişierul **cuvinte.out** toate cuvintele palindromice de lungime L din fişierul de intrare, în ordinea în care apar ele în text.

Un cuvânt este palindromic dacă este simetric faţă de jumătatea cuvântului, exemplu cuvântul **apa**.

23. Pentru două puncte A şi B din plan, puncte date prin coordonatele lor întregi (xa,ya) , (xb,yb) se cere să se verifice dacă punctele A şi B sunt egal depărtate de originea axelor de coordonate. Scrieţi apoi programul care citeşte de la tastatură cele 4 numere întregi şi afişează pe ecran mesajul **DA** în cazul în care A şi B se află la aceeaşi distanţă faţă de punctul O (originea) şi afişează mesajul **NU** în caz contrar.

De exemplu, pentru şirul de valori **24 5 -5 -24** , programul va afisa mesajul **“DA”**.

24. Scrieţi un program care citeşte de la tastatură o propoziţie de cel mult 80 de caractere litere şi spaţii şi apoi elimină spaţiile redundante din propoziţie, adică cele de la începutul şi de la sfârşitul textului, precumşi spaţiile multiple dintre cuvintele propoziţiei.

De exemplu, dacă se citeşte propoziţia : **Afara ninge cu fulgi mari**.

Rezultatul afisat va fi : **Afara ninge cu fulgi mari**.

25. Scrieţi un program care citeşte din fişierul text **date.in** cel mult 100 de numere naturale nenule aflate pe o singură linie, formate din cel mult patru cifre fiecare, separate prin spaţii şi scrie aceste numere în fişierul text **date.out** , în ordine inversă faţă de cea în care au fost citite, pe o singură linie, separate prin spaţii.

De exemplu dacă fişierul **date.in** are următorul conţinut :

1002 2004 1001 1243 5896

atunci fişierul **date.out** va conţine în ordine valorile :

5896 1243 1001 2004 1002.

26. Scrieţi un program care citeşte de la tastatură un număr natural n ($0 \leq n \leq 25$) şi un şir de n numere naturale cu cel mult patru cifre fiecare. Programul va determina şi afişa pe ecran numărul maxim de factori primi care poate să apară în descompunerea în factori primi a numerelor din şir.

De exemplu dacă n=4 iar numerele citite sunt : 24 15 3 29 , atunci rezultatul afişat este 2.

27. Scrieți un program care citește de la tastatură un cuvânt și care scrie în fișierul **cuvant.out** , cuvântul citit sub forma unei elipse, astfel : dacă cuvântul citit este **masa** fișierul va avea următorul conținut

exemplu

xempl

emp

m

emp

xempl

exemplu.

28. Se citește de la tastatură un număr natural n ($1 \leq n \leq 100$) și k număr natural ($1 \leq k \leq n$). Apoi se citesc cele $n \times n$ elemente numere întregi ale matricii a . Scrieți un program care realizează mutarea primelor k coloane ale matricii pe ultimele poziții. Afișați matricea după prelucrare.

De exemplu, pentru $n=4$ și $k=2$ și o matrice formată din elementele :

1 4 6 8	programul va afișa	6 8 1 4
3 1 5 0		5 0 3 1
3 1 5 2		5 2 3 1
4 7 9 2		9 2 4 7

29. Se citesc de la tastatură două numere naturale n și m ($1 \leq n \leq 10$, $1 \leq m \leq 10$) și o matrice a cu n linii și m coloane. Realizați ordonarea elementelor matricii, astfel încât acestea să fie în ordine crescătoare atât pe fiecare linie, cât și pe fiecare coloană. Afișați matricea după prelucrare.

De exemplu, dacă se citesc valorile $n=2$, $m=3$ și matricea

1	4	7
2	6	9
8	3	0

atunci după prelucrare, matricea afișată va fi :

0	1	2
3	4	6
7	8	9

30. Fișierul **numere.in** conține pe prima linie un număr natural n ($0 \leq n \leq 1000000$), iar pe a doua linie n numere reale separate prin câte un spațiu. fiecare număr real este format din cel mult 10 cifre, inclusiv partea zecimală. Scrieți programul care determină și afișează cifrele care nu apar în scrierea nuci unui număr real din fișier. Aceste cifre se vor afișa pe ecran în ordine crescătoare, separate prin câte un spațiu. În cazul în care toate cifrele sunt utilizate în scrierea numerelor din fișier, se va afișa mesajul **NICI UNA**.

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

5

123.47 25.0 -3.69 7.49 -8.42

atunci rezultatul afișat va fi : NICI UNA.