

Biletul nr. 1

Se citesc de la tastatură un număr natural n , apoi o succesiune de n numere naturale cu cel mult 9 cifre fiecare și o cifră k .

Scrieți un program care afișează pe ecran câte numere prime în scrierea cărora apare cifra k , se găsesc în succesiunea de valori citită.

Exemplu

Pentru $n=4$, succesiunea de 4 valori 23 603 122 17 și cifra $k=2$ programul va afișa pe ecran valoarea 1, deoarece doar numărul prim 23 conține cifra 2.

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

1 punct pentru afișarea rezultatului

2 puncte pentru testarea condiției de număr prim

1 punct pentru identificarea tuturor numerelor prime

2 puncte pentru verificarea faptului că cifra k apare într-un număr

1 punct pentru numărarea corectă a valorilor cerute.

Biletul nr. 2

Se citește de la tastatură un număr natural n cu cel mult nouă cifre.

Scrieți un program care determină 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 , urmat de valoarea lui k ; în caz contrar va afișa mesajul NU .

Exemple:

Pentru $n=720$, programul va afișa DA 6

Pentru $n=125$ programul va afișa NU

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

2 puncte pentru afișarea rezultatului

2 puncte pentru structura repetitivă corectă

1 punct pentru structura alternativă corectă

2 puncte pentru identificarea corectă a valorii k

Biletul nr. 3

Se citește de la tastatură un număr natural n , $0 < n < 10000000$.

Scrieți un program care să determine, dacă există, un număr natural care este strict mai mare decât n și care are exact aceleași cifre ca și n .

Programul va afișa pe ecran mesajul `EXISTA`, urmat de numărul determinat, dacă un astfel de număr există; în caz contrar va afișa pe ecran mesajul `NU EXISTA`

Exemple

Pentru $n=16$, se va afișa `EXISTA 61`

Pentru $n=50$ se va afișa `NU EXISTA`

Barem de evaluare:

Se acordă 1 punct din oficiu

1 punct pentru citirea datelor de intrare

1 punct pentru afișarea rezultatului

2 puncte pentru extragerea cifrelor din n

2 puncte pentru construirea unui număr de tipul cerut, dacă există

2 puncte pentru identificarea corectă a existenței numărului

1 punct pentru eficiența rezolvării

Biletul nr. 4

Se citește de la tastatură numărul natural a , $a > 10000$.

Scrieți un program care, folosind descompunerea în factori primi a lui a , decide dacă acesta poate fi exprimat ca produs de două numere prime.

Programul va afișa pe ecran mesajul DA urmat de cele două numere prime identificate, separate prin spațiu (dacă aceste numere există); în caz contrar programul va afișa mesajul NU .

Exemple

Pentru $n=15$, programul va afișa pe ecran $DA \ 3 \ 5$

Pentru $n=16$, programul va afișa pe ecran NU

Barem de evaluare:

Se acordă 1 punct din oficiu

1 punct pentru citirea datelor de intrare

1 punct pentru afișarea rezultatului

3 puncte pentru descompunerea în factori primi a lui n

2 puncte pentru determinarea celor două numere prime (dacă există)

2 puncte pentru identificarea corectă a existenței numerelor

Biletul nr. 5

Fișierul de intrare `valori.txt` conține cel mult 1000 de valori întregi de maxim 9 cifre, aflate pe aceeași linie, despărțite prin câte-un spațiu.

Scrieți un program care să determine și să afișeze pe ecran valoarea maximă și valoarea minimă, împreună cu numărul de apariții ale maximumului, respectiv ale minimumului în fișierul de intrare.

Folosiți o metodă eficientă din punct de vedere al memoriei utilizate și al numărului de operații efectuate.

Exemplu

Dacă fișierul `valori.txt` conține

```
24 3 89 24 1 3 1 1 89
```

Se vor afișa rezultatele:

```
min=1  nr_aparitii=3
```

```
max=89  nr_aparitii=2
```

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

1 punct pentru afișarea rezultatelor

1 punct pentru determinare minim

1 punct pentru determinarea numărului de apariții ale minimumului

1 punct pentru determinare maxim

1 punct pentru determinarea numărului de apariții ale maximumului

1 punct pentru eficiență timp

1 punct pentru eficiență spațiu de memorare

Biletul nr. 6

Fișierul `unu.in` conține pe prima linie valoarea n ($2 \leq n \leq 1\,000\,000$), iar pe a doua linie n numere întregi formate din cel mult 9 cifre, separate prin spații.

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

<code>unu.in</code>	<code>unu.out</code>
10	15 14
-8 14 8 14 15 9 -7 1 4 10	

Barem de evaluare:

- Se acordă 1 punct din oficiu
- 2 puncte pentru citirea datelor de intrare
- 1 punct pentru afișarea rezultatelor
- 2 puncte pentru determinare maximului
- 2 puncte pentru determinarea celui de al doilea maxim
- 1 punct pentru eficiență timp
- 1 punct pentru eficiență spațiu de memorare

Biletul nr. 7

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`.

Exemplu

Dacă se citește șirul `ciocarlie` și caracterul c atunci fișierul `final.out` va conține șirul:

```
ccciocccarlie
```

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

2 puncte pentru afișarea rezultatelor

3 puncte pentru triplarea unei apariții a lui c în s

1 punct pentru triplarea tuturor aparițiilor

1 punct pentru declarații corecte.

Biletul nr. 8

Fișierele `unu.txt` și `doi.txt` conțin pe prima linie numere întregi de cel mult 9 cifre ordonate strict crescător, separate prin spații.

Scrieți un program care creează fișierul `trei.txt` care va conține valorile distincte din cele două fișiere, în ordine strict crescătoare, câte o valoare pe o linie.

Realizați un program eficient din punctul de vedere al memoriei utilizate și al timpului de execuție.

Exemplu

Dacă fișierele conțin:

`unu.txt`

2 4 7 8 42 100 145 200

`doi.txt`

3 7 11 40 100 290

Atunci `trei.txt` va conține valorile:

2

3

4

7

8

11

40

42

145

200

290

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

2 puncte pentru afișarea rezultatelor

2 puncte pentru afișarea valorilor în ordine crescătoare

1 punct pentru valori distincte

1 punct pentru eficiență timp

1 punct pentru eficiență spațiu de memorare

Biletul nr. 9

Se dă șirul recurent definit prin relațiile:

$$x(1)=1$$

$$x(2)=2$$

$$x(n)=2*x(n-2) - x(n-1), \text{ pentru } n>2.$$

Scrieți un program care citește de la tastatură un număr natural n ($0 < n < 10000$) și care afișează în fișierul de ieșire `sir.out` primii n termeni ai șirului recurent dat, în ordine crescătoare, fiecare termen pe o linie.

Exemplu

Pentru $n=5$ primii 5 termeni ai șirului sunt

1 2 0 4 -4

Fișierul `sir.out` va conține

-4

0

1

2

4

Barem de evaluare:

Se acordă 1 punct din oficiu

1 punct pentru citirea datelor de intrare

2 puncte pentru afișarea rezultatelor

3 puncte pentru generarea termenilor șirului

3 puncte pentru afișarea valorilor în ordine crescătoare

Biletul nr. 10

Scrieți un program care citește din fișierul `sir.in` un șir de caractere format din cel mult 50 de litere din alfabetul englez și care afișează pe ecran mesajul ^{DA} dacă șirul este format din număr egal de vocale și consoane, respective mesajul ^{NU} în caz contrar.

Realizați un program eficient din punctul de vedere al spațiului de memorie utilizat.

Exemplu

Dacă fișierul `sir.in` conține șirul

`aaAbbBccee`

Atunci pe ecran va fi afișat mesajul ^{DA}

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

1 punct pentru afișarea rezultatelor

2 puncte pentru identificarea vocalelor/consoanelor

2 puncte pentru numărare corectă

2 puncte pentru eficiență spațiu de memorare

Biletul nr. 11

Scrieți un program care citește din fișierul `siruri.in` 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.

Șirurile se află în fișierul de intrare pe linii diferite.

Exemplu

Dacă primul șir este `"exemplu"`, iar al doilea șir este `"pzyeaxqeemtluss"`, programul trebuie să afișeze mesajul ^{DA} deoarece toate literele primului șir apar în cel de-al doilea șir.

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

1 punct pentru afișarea rezultatelor

2 puncte pentru căutarea unei litere într-un șir

2 puncte pentru căutarea tuturor literelor primului șir

1 punct pentru identificarea situației (DA/NU)

1 punct pentru declarații

Biletul nr. 12

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 ecran, 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.

Exemplu

Dacă fișierul `numere.txt` conține șirul de valori:

10 2

-8 5 1 -3 5 23 7.5 -81 46 -120

Pe ecran va fi afișată valoarea -11 deoarece cele mai mari două valori negative sunt -3 -8.

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

1 punct pentru afișarea rezultatelor

2 puncte identificarea valorilor negative

2 puncte pentru identificarea celor mai mari m valori negative

1 punct pentru însumarea celor m valori

1 punct pentru identificarea situației în care nu există m valori

Biletul nr. 13

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 în fișierul `numere.out`, pe prima linie, separate printr-un singur spațiu.

Exemplu

Pentru $n=3$ și șirul `123 904 500`, se afișează în fișierul `numere.out`
`321 409 5`

Barem de evaluare:

- Se acordă 1 punct din oficiu
- 2 puncte pentru citirea datelor de intrare
- 2 puncte pentru afișarea rezultatelor
- 2 puncte pentru modificarea unui număr
- 2 puncte pentru modificarea tuturor numerelor
- 1 punct pentru eficiență spațiu de memorare

Biletul nr. 14

Scrieți programul C/C++/Pascal care citește două numere naturale a și b ($a \leq b$) cu cel mult 4 cifre fiecare și afișează în fișierul `prime.out` pe prima linie toate numerele prime de maxim 3 cifre din intervalul $[a, b]$. Numerele afișate vor fi separate prin spații.

Realizați un program eficient din punctul de vedere al timpului de execuție

Exemplu

Pentru $a=20$ $b=45$, se afișează

23 29 31 37 41

Barem de evaluare:

Se acordă 1 punct din oficiu

1 punct pentru citirea datelor de intrare

2 puncte pentru afișarea rezultatelor

2 puncte pentru testarea primalității

2 puncte pentru verificarea condiției de maxim 3 cifre

1 punct pentru identificarea tuturor numerelor cu proprietățile cerute

1 punct pentru eficiența algoritmului

Biletul nr. 15

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 $2 * (n+m) - 4$ numere ordonate crescător, șir format din elementele aflate pe chenarul exterior al matricei a . Valorile se vor afișa pe aceeași linie, separate prin spații.

Chenarul exterior este format din prima și ultima linie, respectiv prima și ultima coloană din matrice.

Exemplu

Dacă fișierul de intrare `matrice.txt` conține

```
3 4
1 2 3 4
5 6 7 8
9 1 2 3
```

se va afișa pe ecran șirul:

```
1 1 2 2 3 3 4 5 8 9
```

Realizați un program eficient din punctul de vedere al timpului de execuție și al spațiului de memorie folosit (criteriile de eficiență vor fi considerate în această ordine).

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

1 punct pentru afișarea rezultatelor

2 puncte pentru identificarea valorilor de pe chenarul exterior

2 puncte pentru ordonarea crescătoare a valorilor de pe chenarul exterior

1 punct pentru eficiența timp

1 punct pentru eficiența spațiu de memorare

Biletul nr. 16

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 în fișierul `matrice.out` 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

Barem de evaluare:

Se acordă 1 punct din oficiu

1 punct pentru citirea datelor de intrare

2 puncte pentru afișarea rezultatelor

2 puncte pentru testarea condiției de număr perfect

2 puncte pentru testarea condiției de număr prim

2 puncte pentru construirea corectă a matricii

Biletul nr. 17

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.in` care sunt palindromuri cu exact patru cifre (un număr este palindrom dacă este egal cu inversul său).

Realizați un algoritm eficient din punctul de vedere al spațiului de memorie utilizat.

Exemplu

Dacă fișierul `numere.in` are următorul conținut :

5

1441 5 14 2552 78

Atunci fișierul `numere.out` va conține:

1441

2552

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

2 puncte pentru afișarea rezultatelor

2 puncte pentru testarea condiției de număr palindrom

1 punct pentru testarea condiției de număr cu exact 4 cifre

1 punct pentru identificarea tuturor palindromurilor de exact 4 cifre.

1 punct pentru eficiența algoritmului

Biletul nr. 18

Se citește din fișierul de intrare `intervale.in` un număr natural n , apoi n perechi de câte două numere întregi a și b cu ($a < 0 < b$), numere ce reprezintă capetele unor intervale închise.

Afișați în fișierul `intervale.out` două numere x și y ($x \leq y$) ce reprezintă capetele intervalului **intersecție** al celor n intervale date.

Dacă intersecția celor n intervale date este vidă se vor afișa două valori nule ($0 \ 0$)

Realizați un algoritm eficient din punctul de vedere al spațiului de memorare și al timpului de execuție.

Exemplu

Pentru fișierul de intrare:

```
3
-1 20
-4 9
-7 12
```

Se va afișa soluția

```
-1 9
```

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

2 puncte pentru afișarea rezultatelor

1 punct pentru identificarea cazului în care intersecția este vidă

2 puncte pentru determinarea corectă a intervalului intersecție

1 punct pentru eficiența timp a algoritmului

1 punct pentru eficiența spațiu a algoritmului

Biletul nr. 19

Se citesc de la tastatură două numere naturale n și p ($1 < n < 1000$, $1 < p < 10$).

Să se afișeze pe prima linie în fișierul `nr.out`, cu spații între ele, acele numere naturale mai mici sau egale cu n , care au toate cifrele mai mici sau egale cu p .

Realizați un program eficient din punctul de vedere al timpului de execuție.

Exemplu

Pentru $n=20$ și $p=5$ fișierul de ieșire va conține

1 2 3 4 5 10 11 12 13 14 15

Barem de evaluare:

Se acordă 1 punct din oficiu

1 punct pentru citirea datelor de intrare

2 puncte pentru afișarea rezultatelor

2 puncte pentru verificarea condiției ca toate cifrele să fie $\leq p$

2 puncte pentru determinarea corectă a tuturor numerelor

2 puncte pentru eficiența timp a algoritmului

Biletul nr. 20

Din fișierul de intrare `valori.in` se citește un număr natural n ($1 \leq n \leq 1000$), o succesiune de n valori naturale de maxim 4 cifre și apoi două valori întregi a b , care reprezintă extremitățile intervalului $[a, b]$.

Scrieți un program care să afișeze pe ecran numărul de valori din succesiunea de n valori citită care sunt prime și **nu** se găsesc în intervalul $[a, b]$.

Realizați un program eficient din punctul de vedere al timpului de execuție.

Exemplu

Pentru fișierul de intrare

10

1 4 7 23 90 200 440 500 997 502

10 420

Programul va afișa 3 deoarece numerele prime care nu se află în intervalul $[10, 420]$ sunt 7 23 997

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

1 punct pentru afișarea rezultatelor

2 puncte pentru verificarea condiției de număr prim

1 punct pentru verificarea apartenenței la interval

2 punct numărarea corectă a tuturor valorilor cu proprietățile din enunț

1 punct pentru eficiența timp a algoritmului

2 puncte pentru identificarea cuvintelor

Biletul nr. 22

Pentru două puncte A și B din plan, puncte date prin coordonatele lor întregi (x^a, y^a) , (x^b, y^b) se cere să se verifice dacă punctele A și B sunt egal depărtate de originea axelor de coordonate.

În fișierul de intrare `puncte.in` se găsește pe prima linie un număr natural n , apoi se află n linii, fiecare conținând câte 4 numere întregi x^a y^a x^b y^b , cu semnificația de mai sus.

Scrieți un program care citește perechile de puncte specificate în fișierul de intrare și afișează în fișierul `puncte.out` pentru fiecare pereche citită mesajul `DA` în cazul în care A și B se află la aceeași distanță față de punctul O (originea) și respectiv mesajul `NU` în caz contrar.

Mesajele vor fi scrise pe linii distincte, respectând ordinea din fișierul de intrare.

Realizați un program eficient din punctul de vedere al spațiului de memorie folosit.

Exemplu

Pentru fișierul de intrare

```
3
24 5 -5 -24
1 1 4 4
1 0 -1 0
```

fișierul de ieșire va conține

```
DA
NU
DA
```

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

2 puncte pentru afișarea rezultatelor

3 puncte pentru verificarea condiției ca două puncte să se afle la egală distanță de origine

1 punct pentru identificarea tuturor perechilor de puncte care îndeplinesc această condiție

1 punct pentru eficiența spațiu a algoritmului

Biletul nr. 23

Scrieți un program care citește de la tastatură o propoziție de cel mult 80 de caractere (litere și spații), elimină spațiile redundante din propoziție și afișează pe ecran propoziția obținută.

Numim spații redundante pe cele situate la începutul propoziției, la sfârșitul acesteia, precum și spațiile multiple dintre cuvintele propoziției.

Exemplu

Dacă se citește propoziția

Afara ninge cu fulgi mari

Rezultatul afișat va fi:

Afara ninge cu fulgi mari

Barem de evaluare:

Se acordă 1 punct din oficiu

1 punct pentru citirea datelor de intrare

1 punct pentru afișarea rezultatelor

2 puncte pentru eliminarea spațiilor de la începutul propoziției

2 puncte pentru eliminarea spațiilor de la sfârșitul propoziției

2 puncte pentru eliminarea spațiilor multiple dintre cuvinte

1 punct pentru declarații

Biletul nr. 24

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 patru cifre fiecare, separate prin spații.

Programul va afișa răsturnatul valorilor citite în fișierul text `date.out`, scrise în ordine inversă față de cea în care au fost citite, pe o singură linie, separate prin spații.

Exemplu

Dacă fișierul `date.in` are următorul conținut :

1002 2004 1001 1243 5896

atunci fișierul `date.out` va conține în ordine valorile:

6985 3421 1001 4002 2001

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

2 puncte pentru afișarea rezultatelor

2 puncte pentru determinarea răsturnatului unui număr

2 puncte pentru parcurgerea numerelor în ordine inversă

1 punct pentru declarații

Biletul nr. 25

Scrieți un program care citește din fișierul `nr.in` un număr natural n ($0 \leq n \leq 25$) și un șir de n numere naturale cu cel mult patru cifre fiecare.

Programul va determina și va afișa pe ecran numărul maxim de factori primi care apare în descompunerea în factori primi a unuia dintre numerele din șirul citit.

Realizați un algoritm eficient din punctul de vedere al spațiului de memorie utilizat.

Exemplu

Dacă fișierul de intrare conține:

```
6
24 15 3 720 29 1024
```

Atunci rezultatul afișat este 3, deoarece 720 are 3 factori primi (2, 3, 5).

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

1 puncte pentru afișarea rezultatelor

2 puncte pentru descompunerea unui număr în factori primi

1 punct pentru numărarea corectă a factorilor primi

2 puncte pentru determinarea numărului maxim de factori primi.

1 punct pentru eficiența algoritmului

Biletul nr. 26

Scrieți un program care citește de la tastatură un cuvânt și care scrie în fișierul `cuvant.out` cuvântul citit sub forma unei clepsidre ca în modelul următor, în care cuvântul citit este `exemplu`:

```
exemplu
 xempl
  emp
   m
  emp
 xempl
exemplu
```

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

5 puncte pentru afișarea corectă a rezultatelor

2 puncte pentru declarații

Biletul nr. 27

Din fișierul `matrice.in` se citește un număr natural n ($1 \leq n \leq 100$) și un număr natural k ($1 \leq k \leq n$). Apoi se citesc cele $n \times n$ elemente numere întregi de maxim 4 cifre, reprezentând elementele unei matrice cu n linii și n coloane.

Scrieți un program care realizează mutarea primelor k coloane ale matricei citite pe ultimele poziții, în aceeași ordine. Matricea obținută se va afișa pe ecran.

Realizați un program eficient din punctul de vedere al spațiului de memorie utilizat.

Exemplu

Pentru fișierul de intrare

```
4 2
1 2 3 4
5 6 7 8
9 0 1 2
3 4 5 6
```

Se va afișa matricea

```
3 4 1 2
7 8 5 6
1 2 9 0
5 6 3 4
```

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

1 punct pentru afișarea corectă a rezultatelor

2 puncte pentru mutarea corectă a unei coloane

2 puncte pentru mutarea corectă a tuturor celor k coloane

1 punct pentru declarații

1 punct pentru eficiență

Biletul nr. 28

Se citesc din fișierul de intrare `matrice.in` 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. Elementele matricei sunt numere întregi de maxim 4 cifre.

Afișați pe ecran o matrice cu n linii și m coloane cu elemente ale matricei date, obținută astfel încât parcurgând matricea în ordinea liniilor, iar pe fiecare linie în ordinea coloanelor să se obțină elementele în ordinea crescătoare.

Exemplu

Dacă fișierul de intrare conține

```
2 3
1 4 7
2 6 9
```

Matricea afișată va fi

```
1 2 4
6 7 9
```

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

3 puncte pentru afișarea corectă a rezultatelor

3 puncte pentru ordonarea valorilor

1 punct pentru declarații

Biletul nr. 29

Fișierul de intrare `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 pe ecran cifrele care nu apar în scrierea nici unui număr real din fișierul de intrare. 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 `NICIO CIFRA`

Exemple

Dacă fișierul `numere.in` are următorul conținut :

```
5
123.47 25.0 -3.69 7.49 -8.42
```

atunci rezultatul afișat va fi

```
NICIO CIFRA
```

Dacă fișierul `numere.in` are următorul conținut :

```
5
123.2 25.0 -3.69 2.9 -8.2
```

atunci rezultatul afișat va fi

```
4 7
```

Barem de evaluare:

Se acordă 1 punct din oficiu

2 puncte pentru citirea datelor de intrare

1 puncte pentru afișarea rezultatelor

2 puncte pentru extragerea cifrelor dintr-un număr real

2 puncte pentru identificarea cifrelor care nu apar în niciun număr citit

1 punct pentru ordonarea cifrelor care nu apar

1 punct pentru identificarea cazului în care toate cifrele sunt utilizate