Nr. crt.		**Pentru enunțurile în care nu se precizează altfel, se consideră 2<=n, m<=250, iar elementele matricei sunt numere întregi.**
1.	*	Se consideră un tablou bidimensional cu n linii și m coloane. Realizați un program care afișează coloanele care au elementele ordonate crescător.
2.	*	Se consideră un tablou bidimensional cu n linii și m coloane. Realizați un program care identifică și afișează coloana care are cele mai multe elemente divizibile cu primul element situat pe ea. Dacă sunt mai multe coloane care îndeplinesc această condiție, se va afișa coloana cu numărul de ordine cel mai mare.
3.	*	Se consideră un tablou bidimensional cu n linii și m coloane. Realizați un program care ordonează descrescător fiecare coloană a tabloului.
4.	*	Se consideră un tablou bidimensional cu n linii și m coloane. Realizați un program care calculează suma elementelor de pe fiecare coloană care are primul element număr prim.
5.	*	Se consideră un tablou bidimensional cu n linii și m coloane. Realizați un program care afișează numărul de numere prime de pe fiecare linie a matricii.
6.	*	Se consideră un tablou bidimensional cu n linii și m coloane. Realizați un program care afișează cel mai mare divizor comun al elementelor situate pe fiecare coloană a matricii.
7.	*	Se consideră un tablou bidimensional cu n linii și m coloane. Realizați un program care ordonează crescător elementele impare de pe fiecare linie a matricii, elementele pare nu își vor schimba pozițiile.
8.	*	Se consideră un tablou bidimensional cu n linii și m coloane. Realizați un program care afișează numărul de linii care conțin cele mai multe valori palindrom.
9.	*	Se consideră un tablou bidimensional cu n linii și m coloane. Realizați un program care liniarizează matricea, adică formează un vector cu elementele matricii, parcurse pe coloane.
10.	*	Se consideră un tablou bidimensional cu n linii și m coloane. Realizați un program care interschimbă coloana x cu coloana y, unde x și y sunt citite.
11.	*	Se citește n și m. Se cere să se formeze o matrice cu nxm elemente ale șirului lui Fibonacci, matricea fiind parcursă pe linii.

12.	*	Scrieţi un program C/C++ care citeşte de la tastatură două numere naturale nenule n şi m (2≤m≤10, 2≤n≤10) şi care construieşte în memorie şi apoi afişează o matrice A cu n linii (numerotate de la 1 la n) şi m coloane (numerotate de la 1 la m) cu proprietatea că fiecare element Aij memorează cea mai mică dintre valorile indicilor i şi j (1≤i≤n, 1≤j≤m). Matricea se va afişa pe ecran, câte o linie a matricei pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spaţiu. Exemplu: pentru n=4 şi m=5 se va afişa matricea alăturată. 1 1 1 1 1 1 2 2 2 2 1 2 3 3 3 1 2 3 4 4
13.	*	Scrieţi un program C/C++ care citeşte de la tastatură două numere naturale n şi m (2≤m≤10,2≤n≤10) şi care construieşte în memorie şi apoi afişează o matrice A cu n linii (numerotate de la 1 la n) şi m coloane (numerotate de la 1 la m) cu proprietatea că fiecare element Aij memorează cea mai mare dintre valorile indicilor i şi j (1≤i≤n, 1≤j≤m). Matricea se va afişa pe ecran, câte o linie a matricei pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spaţiu. Exemplu: pentru n=4 şi m=5 se va afişa matricea alăturată. 1 2 3 4 5 2 2 3 4 5 3 3 3 4 5 4 4 4 4 5
14.	*	Scrieţi un program C/C++ care citeşte de la tastatură două numere naturale n şi p (2≤n≤20,1≤p≤20) şi construieşte în memorie un tablou bidimensional cu n linii şi p coloane. Tabloul va fi construit astfel încât, parcurgându-l linie cu linie, de sus în jos şi fiecare linie de la stânga la dreapta, să se obţină şirul primelor n*p pătrate perfecte impare, ordonat strict crescător, ca în exemplu. Tabloul astfel construit va fi afişat pe ecran, fiecare linie a tabloului pe câte o linie a ecranului, cu câte un spaţiu între elementele fiecărei linii. Exemplu: pentru n=2, p=3 se va afişa tabloul alăturat: 1 9 25 49 81 121
15.	*	Se consideră tabloul bidimensional cu n linii şi n coloane ce conţine numere naturale cu cel mult patru cifre fiecare. Scrieţi programul C/C++ care citeşte de la tastatură numărul natural n(2≤n≤23) şi cele n*n elemente ale tabloului şi apoi afişează pe ecran elementele primului pătrat concentric, separate prin câte un spaţiu. Pătratul este parcurs în sensul acelor de ceasornic începând din colţul său stânga-sus, ca în exemplu. Primul pătrat concentric este format din prima şi ultima linie, prima şi ultima coloană a tabloului. Exemplu: pentru n=5 şi tabloul alăturat, se va afişa: 1 2 3 4 5 1 6 2 7 6 5 4 3 7 2 6 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7

16.	*	Scrieţi un program C/C++ care citeşte de la tastatură două numere naturale n şi p ($2\le n\le 20$, $1\le p\le 20$) şi construieşte în memorie un tablou bidimensional cu n linii şi p coloane. Tabloul va fi construit astfel încât, parcurgând matricea linie cu linie de sus în jos şi fiecare linie de la stânga la dreapta, să se obţină şirul primelor n*p pătrate perfecte pare, ordonat strict crescător, ca în exemplu. Tabloul astfel construit va fi afişat pe ecran, fiecare linie a tabloului pe câte o linie a ecranului, cu câte un spaţiu între elementele fiecărei linii. Exemplu: pentru n=2, p=3 programul va afişa tabloul alăturat: 0 4 16 36 64 100
17.	*	Scrieţi un program C/C++ care citeşte de la tastatură două valori naturale nenule m şin ($m\le 10$, $n\le 10$) şi apoi m*n numere naturale nenule cu cel mult 4 cifre fiecare, reprezentând elementele unei matrice cu m linii şi n coloane. Programul determină apoi valorile minime de pe fiecare linie a matricei şi afişează pe ecran cea mai mare valoare dintre aceste minime. Exemplu: pentru m=3, n=5 şi matricea 5 13 7 2 3 9 6 12 9 10 3 6 5 4 7 se afişează pe ecran valoarea 6 (cea mai mică valoare de pe prima linie a matricei este 3, cea mai mică valoare de pe linia a doua este 6, cea mai mică valoare de pe linia a treia este 2. Cea mai mare dintre aceste trei valori este 6).
18.	*	Scrieţi un program C/C++ care citeşte de la tastatură un număr natural n ($2<n<25$) şi apoi construieşte în memorie o matrice cu n linii şi n coloane, numerotate de la 1 la n, ale cărei elemente primesc valori după cum urmează: elementul din linia i şi coloana j primeşte ca valoare ultima cifră a produsului i*j ($1\le i\le n$ şi $1\le j\le n$). Programul va afişa matricea astfel construită pe ecran, câte o linie a matricei pe o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spaţiu. Exemplu: pentru n=4 se va afişa matricea alăturată. 1 2 3 4 2 4 6 8 3 6 9 2 4 8 2 6
19.	*	Scrieţi un program C/C++ care citeşte de la tastatură un număr natural nenul, cu maximum 18cifre, construieşte în memorie şi afişează apoi pe ecran o matrice pătratică, având numărul de linii şi de coloane egal cu numărul de cifre ale numărului dat, completată cu cifrele numărului citit. Astfel, elementele de pe prima coloană a matricei vor fi toate egale cu cifra unităţilor numărului dat, elementele de pe a doua coloană a matricei vor fi toate egale cu cifra zecilor numărului dat, şi aşa mai departe, ca în exemplu. Exemplu: dacă se citeşte numărul 1259, matricea construită va fi cea alăturată. 9 5 2 1 9 5 2 1 9 5 2 1 9 5 2 1

20.	*	Scrieţi un program C/C++ care citeşte de la tastatură un număr natural n (2<n<20) şi construieşte în memorie o matrice cu n linii şi n coloane, numerotate de la 1 la n, în care fiecare element aflat pe chenarul exterior al matricei este egal cu suma dintre indicele liniei şi indicele coloanei pe care se află, iar fiecare dintre celelalte elemente este egal cu suma celor trei “vecini” situaţi în matrice pe linia anterioară. Două elemente din matrice se numesc vecine dacă se găsesc alături pe linie, coloană sau diagonală. Chenarul exterior al unei matrice este format din prima linie, ultima linie, prima coloană şi ultima coloană. Elementele matricei vor fi afişate pe ecran, câte o linie a matricei pe câte o linie a ecranului cu câte un spaţiu între elementele fiecărei linii. Exemplu: pentru n=5 se va obţine matricea alăturată. 2 3 4 5 6 3 9 12 15 7 4 24 36 34 8 5 64 94 78 9 6 7 8 9 10
21.	*	Scrieţi un program C/C++ care citeşte de la tastatură un număr natural n (2≤n≤24) şi construieşte în memorie o matrice cu n linii şi n coloane ale cărei elemente vor primi valori după cum urmează: - elementele aflate pe diagonala principală a matricei vor primi valoarea 0 - elementele de pe prima coloană, cu excepţia celui aflat pe diagonala principală vor primi valoarea n - elementele de pe a doua coloană, cu excepţia celui aflat pe diagonala principală vor primi valoarea n-1 … - elementele de pe ultima coloană, cu excepţia celui aflat pe diagonala principală vor primi valoarea 1 Programul va afişa matricea astfel construită pe ecran, câte o linie a matricei pe câte o linie a ecranului, cu câte un spaţiu între elementele fiecărei linii (ca în exemplu). Exemplu: pentru n=4 se va afişa matricea alăturată. 0 3 2 1 4 0 2 1 4 3 0 1 4 3 2 0
22.	*	Scrieţi un program **C/C++** care citeşte de la tastatură un număr natural **n** (**2<n<16**), construieşte în memorie şi afişează pe ecran o matrice cu **n** linii şi **n** coloane în care elementele de pe cele două diagonale sunt egale cu **0**, elementele care se află deasupra ambelor diagonale sunt egale cu **1**, elementele care se află sub ambele diagonale sunt egale cu **2**, iar restul elementelor sunt egale cu **3**. Elementele matricei vor fi afişate pe ecran, câte o linie a matricei pe câte o linie a ecranului cu câte un spaţiu între elementele fiecărei linii. **Exemplu**: pentru **n=5** se va afişa matricea alăturată. 0 1 1 1 0 3 0 1 0 3 3 3 0 3 3 3 0 2 0 3 0 2 2 2 0

23.	*	Scrieţi un program C/C++ care citeşte de la tastatură un număr natural n (2≤n≤24) şi construieşte în memorie o matrice cu n linii şi n coloane ale cărei elemente vor primi valori după cum urmează: - elementele aflate pe diagonala secundară a matricei vor primi valoarea 0 - elementele de pe prima linie, cu excepţia celui aflat pe diagonala secundară vor primi valoarea n - elementele de pe a doua linie, cu excepţia celui aflat pe diagonala secundară vor primi valoarea n-1 … - elementele de pe ultima linie, cu excepţia celui aflat pe diagonala secundară vor primi valoarea 1 Programul va afişa matricea astfel construită pe ecran, câte o linie a matricei pe câte o linie a ecranului, cu câte un spaţiu între elementele fiecărei linii (ca în exemplu). Exemplu: pentru n=4 se va afişa matricea alăturată. 4 4 4 0 3 3 0 3 2 0 2 2 0 1 1 1
24.	*	Scrieţi un program C/C++ care citeşte de la tastatură un număr natural n (2<n<25) şi apoi construieşte în memorie o matrice cu n linii şi n coloane, numerotate de la 1 la n, ale cărei elemente primesc valori după cum urmează: - elementele aflate pe diagonala secundară sunt toate nule; - elementele de pe coloana i (1≤i≤n), aflate deasupra diagonalei secundare, au valoarea egală cu i; - elementele de pe linia n-i+1 (1≤i≤n), aflate sub diagonala secundară, au valoarea egală cu i. Programul afişează pe ecran matricea construită, câte o linie a matricei pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spaţiu. Exemplu: pentru n=4 se va afişa matricea alăturată. 1 2 3 0 1 2 0 3 1 0 2 2 0 1 1 1
25.	*	Scrieţi un program **C/C++** care citeşte de la tastatură un număr natural **n** (**2<n<20**), construieşte în memorie şi afişează pe ecran o matrice cu **n** linii şi **n** coloane, în care fiecare element de pe diagonala secundară are valoarea **n**, fiecare element aflat deasupra diagonalei secundare este mai mic cu o unitate decât vecinul aflat pe aceeaşi linie în dreapta lui şi fiecare element aflat sub diagonala secundară este mai mare cu o unitate decât vecinul aflat pe aceeaşi linie în stânga lui. Elementele matricei vor fi afişate pe ecran, câte o linie a matricei pe câte o linie a ecranului cu câte un spaţiu între elementele fiecărei linii. **Exemplu:** pentru **n=5** se va afişa matricea alăturată. **1 2 3 4 5** **2 3 4 5 6** **3 4 5 6 7** **4 5 6 7 8** **5 6 7 8 9**

26.	*	Scrieţi un program C/C++ care citeşte de la tastatură un număr natural cu exact 5 cifre şi construieşte în memorie o matrice cu 6 linii şi 6 coloane, numerotate de la 1 la 6, formată astfel: - elementele aflate pe diagonala principală sunt toate nule; - elementele de pe linia 1, aflate deasupra diagonalei principale precum şi elementele de pe coloana 1, aflate sub diagonala principală au toate valoarea egală cu cifra unităţilor numărului citit; - elementele de pe linia 2, aflate deasupra diagonalei principale precum şi elementele de pe coloana 2, aflate sub diagonala principală au toate valoarea egală cu cifra zecilor numărului citit, şi aşa mai departe, ca în exemplu. Matricea astfel construită va fi afişată pe ecran, câte o linie a matricei pe câte o linie a ecranului, elementele de pe aceeaşi linie fiind separate prin câte un spaţiu. Exemplu: dacă se citeşte numărul 28731 matricea construită va fi cea scrisă alăturat. 0 1 1 1 1 1 1 0 3 3 3 3 1 3 0 7 7 7 1 3 7 0 8 8 1 3 7 8 0 2 1 3 7 8 2 0
27.	*	Scrieţi un program C/C++ care citeşte de la tastatură două numere naturale n și m (2<n, m<50), și apoi cele nxm cifre binare ale unui tablou bidimensional. Fiecare linie constituie reprezentarea câte unui număr natural în baza 2. Să se afișeze aceste numere în baza 10. Exemplu, pentru n=3 și m=7 iar matricea este 1 0 0 1 1 0 1 0 1 1 0 1 1 1 1 1 1 0 1 0 1 Se vor afișa valorile 77, 55, 117.
28.	*	Scrieţi un program C/C++ care înmulțește la dreapta o matrice dreptunghiulară de dimensiune (nxm) cu un vector având m elemente, ca în exemplu: $A = \begin{matrix} 1 & 2 & 3 & 5 & 1 \\ 3 & 2 & 11 & 7 & 3 \\ 4 & 6 & 8 & 9 & 12 \end{matrix}$ și X= $\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix}$ atunci AX= $\begin{matrix} 39 \\ 83 \\ 136 \end{matrix}$
29.	*	Un tablou bidimensional A cu m linii şi n coloane (1≤m≤100, 1≤n≤100) conţine pe prima linie numerele 1,2,…,n, iar pe prima coloană numerele 1,2,…,m. Celelalte elemente ale tabloului sunt date de relaţia: Ai,j=Ai-1,j+Ai,j-1. Scrieţi un program C/C++ care citeşte de la tastatură numerele m şi nşi afişează pe ecran ultima cifră a elementului de pe ultima linie şi ultima coloană a tabloului. Exemplu: pentru m=3 şi n=4 se va afişa 5 deoarece elementele tabloului A sunt: 1 2 3 4 2 4 7 11 3 7 14 25

30. *

Să se verifice dacă o matrice pătratică de ordin n este pătrat magic sau nu. O matrice pătratică este pătrat magic dacă sumele de pe fiecare linie, coloană și de pe cele două diagonale sunt egale.

Exemplu, matricea $\begin{matrix} 4 & 3 & 8 \\ 9 & 5 & 1 \\ 2 & 7 & 6 \end{matrix}$ este un pătrat magic.

31. *

Fie A o matrice cu m linii și n coloane, cu numere reale și un vector V cu n componente reale. Să se determine dacă acest vector apare ca linie în matricea A, și în caz afirmativ să se afișeze numărul acestei linii (acestor linii, dacă apare e mai multe ori) sau mesajul NU dacă V nu apare ca linie în matricea A.

Exemplu, pentru matricea $\begin{matrix} 1 & 3 & 6 & 2 & 9 \\ 7 & 4 & 0 & 3 & 8 \\ 5 & 2 & 1 & 47 & 6 \end{matrix}$ și vectorul 5 2 1 47 6 se va afișa valoarea 3.

32. *

Scrieţi un program C/C++ care citeşte de la tastatură un număr natural n (2<n<=10) şi construieşte în memorie o matrice A cu n linii şi n coloane în care toate elementele de pe prima linie, prima şi ultima coloană au valoarea 1 şi oricare alt element Aij din matrice este egal cu suma a 3 elemente situate pe linia i-1: primul aflat pe coloana j-1, al doilea pe coloana j, iar al treilea pe coloana j+1, ca în exemplu. Matricea va fi afişată pe ecran, linie cu linie, numerele de pe aceeaşi linie fiind separate prin câte un spaţiu. Exemplu: pentru n=5 , se afişează matricea alăturată.

1	1	1	1	1
1	3	3	3	1
1	7	9	7	1
1	17	23	17	1
1	41	57	41	1

33. *

Scrieți un program **C/C++** care citeşte de la tastatură un număr natural **n** (**2<=n<=9**) şi elementele unui tablou bidimensional **A** cu **n** linii şi **n** coloane, care memorează numere naturale mai mici decât **10**, şi afişează pe ecran (dacă există), separate prin câte un spaţiu, elementele din matrice care au proprietatea că sunt egale cu produsul celorlalte elemente aflate pe aceeaşi coloană. Dacă nu există astfel de elemente, programul va afişa pe ecran mesajul **NU EXISTA**.
Exemplu: pentru matricea din figura alăturată se afişează, nu neapărat în această ordine, valorile: **4 9** (**4=1*2*2**; **9=3*1*3**).

1	1	2	9
9	2	3	3
8	4	1	1
0	2	9	3

34. *

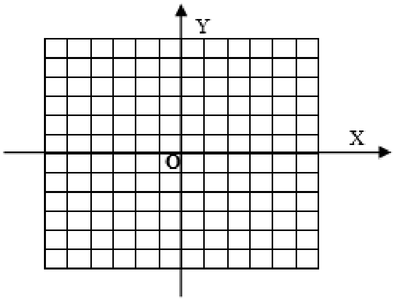
Scrieţi un program C/C++ care citeşte de la tastatură un număr natural n (1≤n≤6) şi elementele unui tablou bidimensional A cu n linii şi n coloane, care memorează numere naturale nenule mai mici decât 100, şi afişează pe ecran produsul numerelor “pivot” pentru matricea A dacă există astfel de numere, altfel va afişa mesajul NU EXISTA. Un număr natural x este “pivot” pentru matricea A dacă înmulţind fiecare element de pe prima coloană cu numărul x, se obţin, în aceeaşi ordine, elementele unei alte coloane din matrice.Exemplu: pentru matricea din figura alăturată se afişează 8.

2	7	4	8	4
1	1	2	4	2
3	6	6	12	3
1	22	2	4	2
5	10	10	20	8

35.	*	Scrieţi programul C/C++ care citeşte de la tastatură un număr natural n (n≤20), construieşte în memorie şi afişează pe ecran, matricea cu n linii şi n coloane, în care se vor memora în ordinea crescătoare a valorii, pe linii şi coloane, primele n^2 numere naturale nenule, pare, care nu sunt divizibile cu 3. Fiecare linie a matricei se va afişa pe câte o linie a ecranului, cu elementele de pe aceeaşi linie separate prin câte un spaţiu. Exemplu: pentru n=4 se va construi şi afişa matricea alăturată. 2 4 8 10 14 16 20 22 26 28 32 34 38 40 44 46
36.	*	Fie un tablou bidimensional cu n linii și m coloane cu elementele numere naturale mai mici decât 10000. Un element din tablou are ca vecini, elementele situate în imediata vecinătate pe verticală și orizontală. Să se identifice un element din matrice care are proprietatea că produsul vecinilor este maxim.
37.	*	Fie un tablou bidimensional pătratic de ordin n cu elemente naturale. Să se elimine linia x și coloana y din matrice.
38.	*	Fie un tablou bidimensional pătratic de ordin n cu elemente naturale. Să se se rotească matricea: - cu 90^0 în sensul trigonometric - cu 180^0 în sensul arcelor de easornic - cu 270^0 în sens trigonometric
39.	*	Fie un tablou bidimensional de dimensiune nxm, cu elemente naturale. a. Să se inverseze elementele egal depărtate de capete, de pe fiecare linie a matricii. b. Să se inverseze elementele egal depărtate de capete, de pe fiecare coloană a matricii.
40.	*	Fie un tablou bidimensional de dimensiune nxn, cu elemente naturale. Să se elimine elementele de pe diagonala principală.
41.	*	Să se rearanjeze elementele unei matrice de dimensiune nxm astfel încât ele să fie ordonate crescător atât pe linii cât și pe coloane.
42.	*	Scrieți un program care afișează elementele unei matrice pătratice nxn, parcurse în spirală, în sensul acelor de ceasornic, plecând de la elementul de pe linia 1 și coloana 1.
43.	*	Scrieți un program care afișează elementele unei matrice pătratice nxn, parcurse în spirală, în sensul trigonometric, plecând de la elementul de pe linia n și coloana n.

44.	*	Fie un tablou bidimensional de n linii și m coloane. Să se șteargă toate liniile care încep cu un element ce se regăsește pe prima linie. Elementele matricei sunt numere naturale, mai mici decât 1000. Exemplu, pentru n=4 și m=4 și matricea 8 6 7 9 7 1 4 5 9 2 7 8 4 2 3 7 se va afișa matricea rămasă 8 7 6 9 4 2 3 7 **Indicație:** Se va folosi un vector de apariții de dimensiune 1000, unde ap[i]=1 dacă valoarea i se găsește pe prima linie a matricei. Ștergerea unei linii x, se va face prin deplasarea tuturor liniilor x+1, x+2, …, n cu o poziție mai sus.
45.	*	Fie un tablou bidimensional cu n linii și m coloane. Să se realizeze un program care inserează în fața fiecărei linii ale cărei elemente sunt ordonate crescător sau descrescător, o nouă linie cu elementele egale cu valoarea maximă de pe linia ordonată. Exemplu, pentru n=2, m=4 și matricea 1 4 5 8 9 8 5 2 se va afișa 8 8 8 8 1 4 5 8 9 9 9 9 9 8 5 7 **Indicație:** inserarea unei linii x se va face prin deplasarea tuturor liniilor începând cu n, n-1, n-2, …, x, cu o poziție mai jos.
46.	*	Se consideră un tablou bidimensional A(n, m) cu elemente întregi. Realizați un program care inversează elementele tabloului, prin intermediul unui vector de nXm elemente. Exemplu, pentru n=3 și m=4, iar matricea 1 2 3 Se va afișa 12 11 10 4 5 6 9 8 7 7 8 9 6 5 4 10 11 12 3 2 1 **Indicație:** se va liniariza matricea, folosind un vector auxiliar X(nxm), fiecare element al matricei A(i,j) se va regăsi în vector pe poziția (i-1)*m+j.
47.	*	Se consideră un tablou bidimensional A(n, m) cu elemente în mulțimea cifrelor 0..9 (2<=n, m<=18). Realizați un program care determină pentru fiecare linie, baza minimă în care cifrele respective pot reprezenta un număr. Considerând că pe fiecare linie este descris n această bază un număr, determinați valoarea obținută la conversia lui în baza 10. Pentru fiecare linie se va afișa baza minimă și valoarea după conversie. Exemplu, pentru n=3 și m=3, iar matricea 1 2 3 se va afișa 4 27 1 0 1 2 5 7 8 9 10 789

48.	*	Se dă o matrice A, pătratică de ordin n. Să se genereze vectorul X având n componente definite astfel: $X[i]=\begin{cases} 1, \text{dacă } \sum_{j=1}^{i} a_{ij} > \sum_{j=1}^{i} a_{ji} \\ 0, \text{în caz contrar} \end{cases}$ Exemplu, pentru matricea 1 2 9 2 2 4 6 7 3 6 4 3 6 1 1 1 5 2 3 9 7 0 10 3 4 Se obține vectorul X=(0, 1, 0, 1, 1).
49.	*	Fie A o matrice pătratică de ordinul n. Să se scrie un program care generează o matrice B pătratică de ordinul n, ale cărei elemente sunt definite prin relația $B_{ij}=(A_{ij}+A_{ji})/2$ş Exemplu, pentru matricea 0 0 14 6 2 2 0 4 0 8 3 6 3 8 7 4 se va obține matricea 0 1 7 4.5 1 2 4 6 7 4 3 6.5 4.5 6 6.5 4
50.	*	Fie un tablou bidimensional pătratic de ordinul n cu elemente numere naturale. Se consideră un traseu ce pleacă din matrice de pe linia x și coloana y. Direcția de mișcare este dată de un șir de p caractere N, E, V, S, care indică direcția de deplasare. Determinați suma elementelor situate pe drum. Elementul de start aparține drumului. Exemplu, pentru n=4, x=3 și y=2, p=6, traseul N N E S V V și tabloul 0 9 5 8 3 2 1 5 3 8 3 0 0 2 6 2 Se va afișa 30 (8+2+9+5+1+2+3).
51.	**	O metodă foarte simplă de generare a unui pătrat magic constă în completarea pe rând cu valorile 1, 2, 3, …, n^2, astfel: - Se plasează 1 în centrul ultimei coloane - Se merge în linie oblică, dreapta jos, cu umărul următor; dacă se iese din pătrat prin partea dreaptă, se merge în partea opusă, în stînga liniei unde trebuia depus numărul, iar dacă se iese prin partea de jos a pătratului, se merge în partea de sus a coloanei unde trebuia depus numărul; - După ce se completează un grup de n numere, se merge cu o căsuță spre stînga, pe aceeași linie,, pentru a se repeta apoi pasul 2 și a genera următorul grup de n numere. Să se scrie programul care implementează acest algoritm. Exemplu, pentru n=5 se obține în final următorul pătrat magic: 11 10 4 23 17 18 12 6 5 24 25 19 13 7 1 2 21 20 14 8 9 3 22 16 15

<table>
<tr>
<td>52.</td>
<td>**</td>
<td>O reţea este formată din noduri situate în punctele de coordonate întregi, fiecare nod fiind unit prin bare paralele cu axele de coordinate, de cele 4 noduri vecine. Un păianjen este plasat iniţial în originea sistemului de coordonate. La fiecare secundă, păianjenul se poate deplasa din nodul în care se află în unul dintre cele 4 noduri vecine.


Scrieţi un program care să determine în câte moduri se poate deplasa păianjenul din poziţia iniţială, într-o poziţie finală dată, în timpul cel mai scurt.
Fişierul de intrare spider.in conţine pe o singură linie abscisa şi ordonata punctului final, separate prin spaţiu:
x y
în fişierul de ieşire spider.out se va afişa pe prima linie numărul de moduri determinat, nr.
0<x, y≤80
exemplul 1
<table>
<tr><td>spider.in</td><td>spider.out</td></tr>
<tr><td>1 2</td><td>3</td></tr>
</table>
exemplul 2
<table>
<tr><td>spider.in</td><td>spider.out</td></tr>
<tr><td>2 3</td><td>10</td></tr>
</table>
</td>
</tr>
<tr>
<td>53.</td>
<td>**</td>
<td>O livadă este împărțită în nxm zone. În fiecare zonă crește câte un pom. Din fiecare pom cade pe jos o cantitate de fructe. În zona stânga sus se află un arici care vrea sa ajungă în zona dreapta jos. Ariciul se poate deplasa doar pe două direcții: în jos sau spre dreapta.
Determinați cantitatea maximă de fructe pe care le poate aduna ariciul prin deplasarea din poziția initială în cea dorită.
Citirea se face din fișierul arici.in care conține pe prima linie dimensiunile livezii, adică n și m, și apoi cantitatea de fructe din fiecare dintre cele nxm zone.
Afișarea cantității maxime de fructe se va face în fișierul arici.out.
Exemplu:

arici.in
3 3
0 4 1
0 1 1
1 0 1

arici.out
7</td>
</tr>
</table>

54.	**	Fie un tablou bidimensional pătratic de ordinul n cu elemente numere naturale. Se consideră un traseu ce pleacă din matrice de pe linia x și coloana y. Direcția de mișcare este dată de un șir de p caractere N, E, V, S, care indică direcția de deplasare. Determinați elementele prin care s-a trecut de cele mai multe ori. Din fișier se citesc n, x, y, p și traseul urmat. Pentru fiecare element din soluție se va afișa linia și coloana pe care este situat. Exemplu, pentru n=4, x=3, y=2, p=8 și traseul N N E S V S S V Se va afișa 2 2 3 2
55.	**	Se consideră o tablă de șah cu n linii și m coloane, pe care sunt situați pioni. Pionii sunt codificați la citire prin valoarea 1. Regina adversă trebuie plasată într-un punct al tablei astfel încât pe cele două diagonale pe care le atacă, să se afle cât mai mulți pioni. Deteminați poziția (linia și coloana) pe care se va așeza regina și numărul de pioni pe care aceasta le atacă. Exempu, pentru n=5 și m=4 și tabloul 1 1 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 0 1 0 Se va afișa 4 2 5 (linia 4, coloana 2 și 5 pioni atacați).
56.	**	Fie un tablou bidimensional cu n linii și m coloane, cu elemente întregi. Din punctul de pe linia x și coloana y se poate părăsi tabloul mergând numai pe orizontală sau pe verticală, dar numai dacă elementul a[x][y] este negativ. Pe traseu urmat nu trebuie să se întâlnească alt element negativ. Să se determine linia și coloana de unde se poate părăsi matricea, astfel încât suma elementelor întâlnite pe traseu să fie minimă. Afișați linia ți coloana punctului de start și suma elementelor de pe drum. De exemplu, pentru n=4 și m=6 și tabloul 50 60 90 50 60 60 40 30 -9 -3 40 70 40 -1 -8 5 2 1 80 80 80 80 90 20 Se va afișa 3 3 8 (linia 3 coloana 3 suma 8)
57.	**	a) Se citesc două numere naturale n și m (cel mult 1000). Să se construiască o matrice cu n linii și cu m coloane care să conțina numerele de la 1 la m în zigzag, ca în exemplul de mai jos, restul elementelor din matrice fiind 0. Exemplu: pentru n=4 și m=11 matricea este 1 0 0 0 0 0 7 0 0 0 0 0 2 0 0 0 6 0 8 0 0 0 0 0 3 0 5 0 0 0 9 0 11 0 0 0 4 0 0 0 0 0 10 0 b) b) Se continuă plasarea numerelor prin revenirea de la ultima spre prima coloană după aceeași regulă. Exemplu: pentru n=6 și m=9 matricea este 1 0 0 0 0 0 11 0 0 0 2 0 0 0 12 0 10 0 0 0 3 0 13 0 0 0 9 0 0 0 14 0 0 0 8 0 17 0 15 0 5 0 7 0 0 0 16 0 0 0 6 0 0 0

<table>
<tr><td>58.</td><td>**</td><td>Se dau n(n+1)/2 numere naturale aranjate într-un triunghi format din elementele de sub și de pe diagonala principală ale unei matrice patratice de ordin n.
Se calculează sume pornind din elementul de pe prima linie prin deplasări spre vecinii de sub și din dreapta. Găsiți suma maximă care se poate calcula astfel și care sunt valorile din care se obține suma maximă.
Exemplu:
n=4
2
3 5
6 3 4
5 6 1 4
suma maximă este 17
si se obține din valorile 2 3 6 6</td></tr>
<tr><td>59.</td><td>**</td><td>O râmă se mişcă într-o zonă pătratică, intrând şi ieşind din pământ. Ea înaintează spre centrul zonei, plecând de la suprafață din (1,1) și mergând paralel cu cele patru laturi, fără sssăăă treacă de două ori printr-un loc., descriind astfel o spirală, ordinea de deplasare este exemplificată astfel:
1 2 3 4
12 13 14 5
11 16 15 6
10 9 8 7
Deplasarea se face alternativ, fie la suprafaţa pământului, fie pe sub pământ. La întâlnirea unei gropi pe traseu râma va intra în groapă dacă deplasarea se făcea la suprafaţă, sau va ieşi la suprafaţă dacă deplasarea se făcea pe sub pământ. Pe hartă gropie sunt codificate cu valoarea 0, restul valorilor fiind egale cu 1. La coordonatele (1,1) nu poate exista o groapă.
Realizaţi un program prin care sunt identificate gropile pe care râma le-a folosit pentru a ieşi la suprafaţă. Acestea vor fi enumerate în ordinea întâlnirii lor pe traseu.
În fişierul rama.in se găseşte pe prima linie un număr natural n, iar pe următoarele n linii este descrisă harta prin cifre binare, despărţite printr-un spaţiu. În fişierul rama.out se va găsi, pe prima linie, numărul de gropi identificate, iar pe următoarele linii câte o pereche de numere naturale, separate printr-un spaţiu, reprezentând coordonatele gropilor, în ordine întâlnirii lor pe traseu.
<table><tr><td>rama.in</td><td>rama.out</td><td>rama.in</td><td>rama.out</td></tr><tr><td>4</td><td></td><td>4</td><td></td></tr><tr><td>1 1 0 1</td><td>2</td><td>1 1 1 0</td><td>3</td></tr><tr><td>1 1 1 1</td><td>3 4</td><td>1 0 1 0</td><td>2 4</td></tr><tr><td>1 1 1 0</td><td>4 1</td><td>1 1 0 1</td><td>4 2</td></tr><tr><td>0 0 1 1</td><td></td><td>0 0 0 1</td><td>2 2</td></tr></table></td></tr>
<tr><td>60.</td><td>**</td><td>Se dă un şi de numere întregi de dimensiune nxn. Să se genereze o matrice A(nxn) astfel încât elementele matricei să reprezinte elementele vectorului X scrise în următoarea ordine:
X[1] X[2] X[5] ⋮
X[4] X[3] X[6] ⋮
X[9] X[8] X[7] ⋮
… … …
De exemplu, pentru n=3 şi X=(1, 5, 3, 9, 4, 7, 6, 10, 3) se obţine matricea
1 5 4
9 3 7
3 10 6</td></tr>
</table>

61.	**	Se dă un și de numere întregi de dimensiune nxn. Să se genereze o matrice A(nxn) astfel încât elementele matricei să reprezinte elementele vectorului X scrise în următoarea ordine: X[1] X[2] X[4] X[7] ⋮ X[3] X[5] X[8] ⋮ X[6] X[9] ⋮ X[10] … … … De exemplu, pentru n=3 și X=(1, 5, 3, 9, 4, 7, 6, 10, 3) se obține matricea 1 5 9 3 4 6 7 10 3
62.	**	Să se completeze în spirală o matrice cu n linii și n coloane, cu primele n^2 numere întregi cu cel puțin 2 cifre, cu proprietatea că suma cifrelor lor este divizibilă cu 5. Exemplu, pentru n=4 se obține matricea 14 19 23 28 69 73 78 32 64 87 82 37 55 50 46 41
63.	**	Se dă un tablou bidimensional A cu n linii și n coloane, având componente binare care codifică relațiile de prietenie între elevii unei clase. Un element A[i][j] este 1 dacă i și j sunt prieteni. Se cere să se afișeze copiii care au cei mai mulți prieteni, știind că relația de prietenie este reciprocă. Exemplu, dacă relațiile dintre n=5 copii, numerotați 1, 2, 3, 4, 5 se codifică prin următoarea matrice 0 1 0 1 1 1 0 1 0 0 0 1 0 0 1 1 0 0 0 1 1 0 1 1 0 Atunci elevii 1 și 5 au cei mai mulți prieteni (3 prieteni)
64.	**	Se dă o matrice cu n linii și m coloane reprezentând o populație, a_{ij}=1 reprezintă o căsuță ocupată de un membru al popolației, iar a_{ij}=0 o căsuță liberă. Să se determine factorul de aglomerare al populației, care se definește ca fiind numărul maxim de vecini ai unui membru al populației. Exemplu, pentru următoarea configurație (n=4, m=6) 0 1 1 0 1 0 0 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 0 Gradul de aglomerare este 6.
65.		O matrice pătratică A conține doar elemente 0 și 1. Să se afișeze pozițiile celor mai îndepărtate două elemente de 1 (între ele afșându-se doar valori de 0), distanța dintre două elemente fiind calculată ca diferența numerelor de ordine ale acestor elemente obținute prin liniarizarea pe linii a matricei. (Nu se va realiza efectiv liniarizarea matricei.) Exemplu: în matricea 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 Cele mai îndepărtate elemente de 1 sunt a[3][5] și a[5][2] între ele găsindu-se 8 elemente de 0.
66.	**	Se dă o matrice de numere reale, avînd m linii și n coloane. Să se schimbe între ele liniile matricei astfel încât prima coloană să devină ordonată crescător. De exemplu matrricea 4 5 4 1 9 7 1 9 7 devine după ordonare 2 4 3 2 4 3 4 5 4

67.	**	Se consideră o matrice A de dimensiune nxm. Să se rearanjeze elementele în matrice astfel încât ele să apară în ordine crescătoare atât pe linii cât și pe coloane. Exemplu, dacă matricea inițială este 2 98 36 1 7 14 1 14 7 se poate obține matricea 2 18 55 (nu este unica soluție!!) 55 18 12 12 36 98
68.	**	Se consideră o matrice pătratică de dimensiune nxn. Să se rearanjeze elementele matricei astfel încât ele să apară în ordine crescătoare pe diagonala principală, prin interschimbarea liniilor și coloanelor. Astfel, pentru matricea 4 5 4 3 2 4 1 9 7 se va obține matricea 4 4 5 2 4 3 7 1 9
69.	**	Se citesc două matrice A(nxm) și B(mxp). Să se afişeze matricea C(nxp) ce reprezintă produsul matricelor AxB (**Indicație:** c[i][j]=c[i][j]+a[i][k]*b[k][j], unde k=$\overline{1..m}$) Exemplu $\begin{bmatrix}1 & 0\\ 4 & 2\\ 1 & 1\end{bmatrix} x \begin{bmatrix}1 & 2 & 1\\ 1 & 0 & 3\end{bmatrix} = \begin{bmatrix}1 & 2 & 1\\ 6 & 8 & 10\\ 2 & 2 & 4\end{bmatrix}$ unde n=3, m=2, p=3
70.	**	Se dă un tablou bidimensional cu m linii și n coloane (1≤m, n ≤50) având componente cifre zecimale. Fiecare linie a matricei reprezintă cifrele câte unui număr natural. Se cere să se afișeze cifrele sumei celor m numere date prin tabloul descris mai sus. Exemplu: pentru m=3 și n=7 și matricea 9 0 9 8 3 4 9 3 5 9 9 8 7 9 4 9 3 6 7 9 0 Se va afișa valoarea 17635018
71.	***	**(Algoritmul lui Lee)** Pe o tablă dreptunghiulară cu n linii și m coloane, există zone libere marcate cu 0 și zone cu obstacole, marcate cu 1. Știind că pe tablă se află un șoricel la poziția xi, yi, și o bucată de brânză la poziția xf, yf, să se găsească traseul minim pe care trebuie să îl străbată șoricelul până la brânză. Acesta se poate mișca pe tablă în oricare zonă alăturată (pe linie, pe coloană sau pe diagonală) poziției curente, unde nu se află un obstacol. Din fișierul mouse.in se vor citi, în ordine, numerele n, m xi, yi, xf, yf, iar de e următoarele n linii, codificarea tablei. Traseul determinat se va afișa pe ecran, pe o singură linie, în formatul din exemplu. Exemplu Mouse.in 4 6 3 1 1 6 0 0 0 0 0 1 0 0 1 1 0 0 0 0 1 1 0 0 0 1 0 0 0 0 Se va afișa (3,1)(2,2)(1,3)(1,4)(2,5)(1,6)

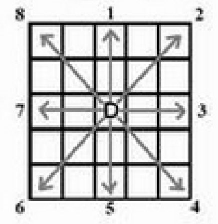
72. ***

Robinson are un teren de forma pătrată, ca un tablou bidimensional, cu latura de m unităţi (cu m^2 parcele). O parcelă este un pătrat elementar al tabloului. Liniile şi coloanele sunt numerotate de la 1 la m: liniile de sus în jos, iar coloanele de la stânga la dreapta.
El a semănat grâu şi s-a rugat să aibă o recoltă bogată. Ruga i-a fost ascultată şi grâul a răsărit astfel: în parcelele din prima linie, recoltele au fost de la stânga spre dreapta: n, n+1, n+2, … boabe de grâu şi în parcelele de pe prima coloană, recoltele au fost de sus în jos: n, n+1, n+2, … boabe de grâu.
Apoi, dacă parcurgem celelalte parcele linie cu linie începând cu a doua linie, iar în cadrul unei linii, începând cu coloana a doua, atunci recolta din linia i şi coloana j a fost egală cu suma recoltelor de pe poziţiile (i-1,j) şi (i,j-1).
Dacă aceste sume depăşesc 999, atunci ele vor fi înlocuite cu numerele formate din ultimele 3 cifre ale sumei respective (de exemplu, dacă suma=1234, se va reţine numărul 234).
Odată cu împlinirea rugii, Robinson a avut un vis în care, pentru a avea noroc şi anul viitor, i se cerea ca în prima zi să culeagă grâul astfel: să plece de la o poziţie dată (linia l şi coloana c), de unde va culege toate boabele de pe acea poziţie. Apoi, el va calcula restul împărţirii la 4 a numărului de boabe de pe acea poziţie. Poziţia următoare pentru cules va fi cea vecină din Nord, dacă restul este 0, cea vecină din Est, dacă restul este 1, cea vecină din Sud, dacă restul este 2, sau cea vecină din Vest, dacă restul este 3.
Drumul acesta pe care l-a visat se va opri fie când la poziţia următoare este în afara terenului, fie când poziţia următoare este una de pe care s-a cules deja recolta.
Scrieţi un program care să citească numerele m, n, l şi c şi care să determine:
a) recoltele de pe fiecare parcelă;
b) succesiunea parcelelor vizitate, în ordinea în care s-a cules grâul în drumul visat;
De pe prima linie a fişierului de intrare robinson.in se citesc numerele m n l c în această ordine, separate de câte un spaţiu; l şi cindică o poziţie corectă din tablou.
Pe prima linie a fişierului de ieşire robinson.out se va afişa valoarea a[m][m]. Următoarele linii vor conţine câte două numere naturale separate de un spaţiu indicând coordonatele fiecărei parcele din drumul parcurs: primul număr indică linia iar al doilea număr coloana parcelei vizitate.
m, n, l, c sunt numere naturale, 1<= m <=20, 1<= n <=100.

robinson.in	robinson.out	Explicaţii
4 55 1 3	130 1 3 1 4 2 4 2 3	a) m=4, adică tabloul pătrat are latura de 4 unităţi. n=55, deci prima linie este: 55, 56, 57 şi 58. La fel şi prima coloană: 55, 56, 57, 58, de sus în jos. Apoi linia a doua se completează astfel: a[2,2]=a[1,2]+a[2,1]=56+56=112, apoi a[2,3] = a[1,3]+a[2,2]=57+112=169 ; apoi a[2,4]=a[1,4]+a[2,3] etc. Linia a treia, se va completa astfel: a[3,2]=a[2,2]+a[3,1]; apoi a[3,3]=a[2,3]+ a[3,2], apoi a[3,4]= a[2,4]+ a[3,3] etc. Atentie: a[4,4]=a[3,4]+a[4,3]=565+565=1130 şi se reţine a[4,4]=130, adică numărul format din ultimele 3 cifre ale lui 1130. b) Poziţia de plecare: a[1;3]=57, cu rest 1, deci direcţia este Est. Aici avem a[1,4]=58, care are rest 2, deci direcţia este Sud. Aici a[2,4]=227, cu rest 3, deci direcţia este Vest, unde avem a[2,3]=169. Acesta are restul 1 şi direcţia este Est, şi ar trebui să se revină pe poziţia (2,4) pe care a mai fost. S-au afişat poziţiile (1;3), (1;4), (2,4) şi (2;3). Dacă în loc de l=1 şi c=3 am fi avut l=3 şi c=4, atunci a[3,4] = 565, cu rest 1, deci direcţia Est şi ar trebui să iasă din teren. Drumul acesta ar avea un pas.

73. ***

Fie o tablă de şah, de dimensiune nxn. Pe suprafaţa de joc sunt aşezate D dame în D pătrate albe distincte, ocupându-le. Într-un pătrat alb poate fi aşezată o singură damă, iar într-un pătrat gri nu poate fi aşezată nicio damă. Poziţia unei dame pe suprafaţa de joc este dată de poziţia pătratului alb în care este aşezată dama. Damele pot accesa orice pătrat alb neocupat situat pe direcţiile: verticală, orizontală sau diagonală, numerotate de la 1 la 8 în figura de mai jos).



Accesul pe o direcţie se face trecând din pătrat alb în pătrat alb (doar pătrate albe neocupate) până la întâlnirea unui pătrat gri sau a unui pătrat alb ocupat de o altă damă sau până la terminarea suprafeţei de joc. Numim pătrat accesibil orice pătrat alb neocupat (de pe suprafaţa de joc) care ar putea fi accesat de cel puţin una din cele D dame.

Scrieţi un program care să citească numerele naturale N D K, poziţiile damelor şi ale pătratelor gri pe suprafaţa de joc şi care să determine:

a) numărul maxim M de pătrate albe conţinute de un rând al suprafeţei de joc;

b) numărul P de pătrate accesibile de pe suprafaţa de joc.

Fişierul de intrare betasah.in conţine:

- pe prima linie cele trei numere naturale N D K, separate prin câte un spaţiu, cu semnificaţia din enunţ;
- pe linia i+1 două numere naturale nenule x_i y_i, separate printr-un singur spaţiu, reprezentând poziţia damei i pe suprafaţa de joc (rândul x_i şi coloana y_i), pentru i=1,2,3,…,D;
- pe linia D+1+j două numere naturale nenule z_j t_j, separate printr-un singur spaţiu, reprezentând poziţia pătratului gri j pe suprafaţa de joc (rândul z_j şi coloana t_j), pentru j=1,2,3,…,K.

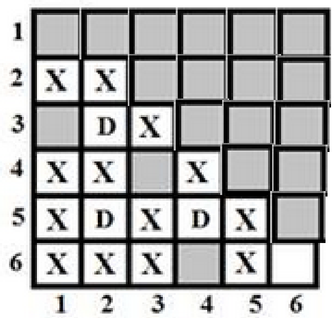
Fişierul de ieşire betasah.out va conţine pe prima linie numărul natural M şi pe a doua linie numărul natural P, cu semnificaţia din enunţ.

Betasah.in

6 3 19
3 2
5 2
5 4
3 1
4 3
6 4
1 1
1 2
1 3
1 4
1 5
1 6
2 3
2 4
2 5
2 6
3 4
3 5
3 6
4 5
4 6
5 6

Explicaţii

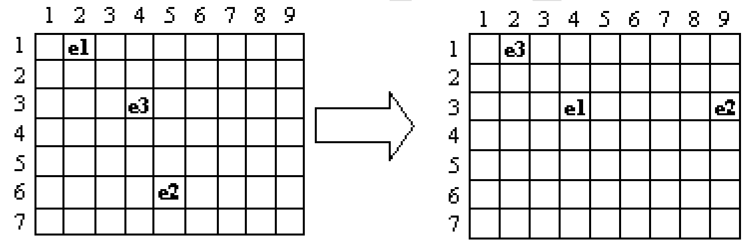
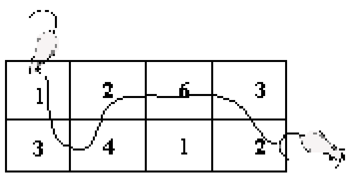
N=6, D=3, K=4.
Rândurile 5 şi 6 conţin numărul maxim M=5 de pătrate albe.
Numărul de pătrate accesibile de pe suprafaţa de joc este P=13.
În desenul alăturat corespunzător suprafeţei date, cele 13 pătrate accesibile sunt marcate cu X.
Astfel, pe prima linie a fişierului `betasah.out` se va scrie numărul 5, iar pe a doua linie a fişierului se va scrie numărul 13.



În fişierul batasah.out se va afişa

5
13

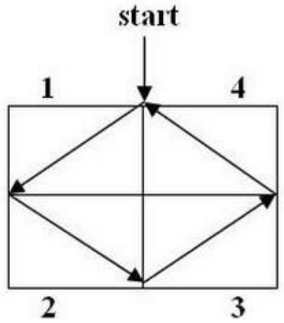
<table>
<tr>
<td>74.</td>
<td>***</td>
<td>Se consideră harta universului ca fiind o matrice cu 250 de linii şi 250 de coloane. În fiecare celulă se găseşte o aşa numită poartă stelară, iar în anumite celule se găsesc echipaje ale porţii stelare. La o deplasare, un echipaj se poate deplasa din locul în care se află în oricare alt loc în care se găseşte o a doua poartă, în cazul nostru în orice altă poziţie din matrice. Nu se permite situarea simultană a mai mult de un echipaj într-o celulă. La un moment dat un singur echipaj se poate deplasa de la o poartă stelară la alta.
Dându-se un număr <i>p</i> (1<<i>p</i><5000) de echipaje, pentru fiecare echipaj fiind precizate poziţia iniţială şi poziţia finală, determinaţi numărul minim de deplasări necesare pentru ca toate echipajele să ajungă din poziţia iniţială în cea finală.
Se citesc din fisierul text poarta.in în următorul format:
– pe prima linie numărul natural p reprezentând numărul echipaje,
– pe următoarele p linii câte 4 numere naturale, primele două reprezentând coordonatele poziţiei iniţiale a unui echipaj (linie coloană), următoarele două reprezentând coordonatele poziţiei finale a aceluiaşi echipaj (linie coloană).
Pe prima linie a fişierului text poarta.out se scrie un singur număr reprezentând numărul minim de deplasări necesar.
poarta.in
3
1 2 3 4
6 5 3 9
3 4 1 2

poarta.out
4

</td>
</tr>
<tr>
<td>75.</td>
<td>***</td>
<td>Un experiment urmăreşte comportarea unui şoricel pus într-o cutie dreptunghiulară, împărţită în m×n cămăruţe egale de formă pătrată. Fiecare cămăruţă conţine o anumită cantitate de hrană. Şoricelul trebuie să pornească din colţul (1,1) al cutiei şi să ajungă în colţul opus, mâncând cât mai multă hrană. El poate trece dintr-o cameră în una alăturată (două camere sunt alăturate dacă au un perete comun), mănâncă toată hrana din cămăruţă atunci când intră şi nu intră niciodată într-o cameră fără hrană. Stabiliţi care este cantitatea maximă de hrană pe care o poate mânca şi traseul pe care îl poate urma pentru a culege această cantitate maximă.
Fişierul de intrare mouse.in conţine pe prima linie două numere m şi n reprezentând numărul de linii respectiv numărul de coloane ale cutiei, iar pe următoarele m linii cele m×n numere reprezentând cantitatea de hrană existentă în fiecare cămăruţă, câte n numere pe fiecare linie, separate prin spaţii. Toate valorile din fişier sunt numere naturale între 1 şi 100.
În fişierul de ieşire mouse.out se vor scrie pe prima linie două numere separate printr-un spaţiu: numărul de cămăruţe vizitate şi cantitatea de hrană maximă culeasă. Pe următoarele linii se va scrie un traseu posibil pentru cantitatea dată, sub formă de perechi de numere (linie coloană) începând cu 1 1 şi terminând cu m n.
mouse.in
2 4
1 2 6 3
3 4 1 2
mouse.out
7 21
1 1
2 1
2 2
1 2
1 3
1 4
2 4

</td>
</tr>
</table>

76. ***

În profesia sa de hoţ, Ionel a întâlnit o nouă provocare: la un nou “loc de muncă”, în locul seifului clasic, a întâlnit un afişaj în formă pătrată conţinând diverse cifre dispuse orizontal şi vertical, în număr egal pe linii şi coloane şi o tastatură pentru introducerea parolei care deschide seiful.

Pentru a forma această parolă se împarte afişajul în 4 cadrane, apoi se preiau cifrele situate pe diagonalele acestor cadrane, la parcurgerea lor în sensul invers arcelor de ceas, ca în imaginea alăturată.



Parola se obţine rotind imaginar la stânga cu un număr precizat k de poziţii, şirul cifrelor determinat anterior.

Cunoscând numărul natural n, apoi cifrele inscripţionate pe tastele de pe cele n linii şi n coloane, se cere să se afişeze şirul de taste ce formează parola ce deschide seiful. Parola va conţine şirul de cifre pornind de la poziţia de start, după rotirea acestora la stânga cu kpoziţii.

Fişierul seif1.in conţine pe prima linie se găsesc scrise două numere separate prin spaţiu: numarul natural n, reprezentând numărul de linii şi coloane ale afişajului şi numărul natural k reprezentând numărul de rotiri la stânga ale şirului de cifre selectate pentru a obţine parola seifului. Pe următoarele n linii se găsesc scrise câte n cifre separate prin câte un spaţiu, reprezentând afişajul.

Fişierul seif1.out va conţine pe o singură linie despărţite printr-un spaţiu, şirul de cifre ce formează parola seifului.

- $3 < n \leq 100$
- n număr par
- $0 < k < 1000$

seif1.in	seif1.out	Explicaţii
6 3 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6	9 6 3 4 9 4 8 1 4 3 8 3	6 3 1 2 $\boxed{3}$ $\boxed{4}$ 5 6 7 $\boxed{8}$ 9 0 $\boxed{1}$ 2 $\boxed{3}$ 4 5 6 7 $\boxed{8}$ $\boxed{9}$ 0 1 2 3 $\boxed{4}$ 5 $\boxed{6}$ 7 8 $\boxed{9}$ 0 1 2 $\boxed{3}$ $\boxed{4}$ 5 6 Iniţial cheia seifului este formată din cifrele: 3 8 3 9 6 3 4 9 4 8 1 4 După 3 rotiri la stânga cheia seifului va fi formată din cifrele: 9 6 3 4 9 4 8 1 4 3 8 3.

77. ***

Un fermier deţine un teren dreptunghiular format din N linii şi M coloane. Terenul său este împărţit în N*M pătrate, fiecare pătrat de teren fiind identificat prin două coordonate (linie, coloană). Din păcate pentru fermier, unele pătrate sunt compromise pentru agricultură, fiind complet acoperite cu pietre. Cunoaştem coordonatele pătratelor care sunt acoperite. Fermierul cultivă întreaga suprafaţă liberă cu grâu şi porumb astfel: începe din colţul (1,1) cu grâu, cultivă linia 1, apoi linia 2 şi aşa mai departe, până ajunge la pietre. Trece de poziţia sau poziţiile cu pietre şi continuă cu porumb o zonă liberă până ajunge din nou la pietre, apoi cultivă grâu, apoi porumb, şi tot aşa, până termină de cultivat tot terenul liber. Fiecare linie este cultivată de la stânga la dreapta (linia i începe cu pătratul (i,1)). Dacă terenul are pietre la început (colţul (1,1) este ocupat) fermierul trece de pietre şi începe cultivarea pe primul pătrat liber cu grâu.
Scrieţi un program care determină câte pătrate sunt cultivate cu grâu.
Fişierul fermier.in conţine pe prima linie numerele naturale N, M, P (cu semnificaţia N = numărul de linii ale terenului, M = numărul de coloane ale terenului, P = numărul de pătrate de teren compromise) şi pe următoarele P linii câte două numere naturale x_i şi y_i (i = 1, 2, … P) ce reprezintă linia, respectiv coloana câte unui pătrat acoperit cu pietre.
Fişierul fermier.out va conţine pe prima linie numărul de pătrate cultivate cu grâu.

Restricţii

- $1 < N,M \leq 40000$;
- $1 < P \leq 4000$;
- $1 \leq xi \leq N$; $1 \leq yi \leq M$

fermier.in	fermier.out
7 5 8 3 3 3 4 1 1 7 1 7 2 6 5 5 2 5 1	18

Explicaţie

Terenul fermierului este:

*	G	G	G	G
G	G	G	G	G
G	G	*	*	P
P	P	P	P	P
*	*	G	G	G
G	G	G	G	*
*	*	P	P	P

unde * = pietre, G = grâu, P = porumb.
Sunt 18 pătrate cultivate cu grâu.

78. ***

Legenda spune că după înfrângerea lui Montezuma, conchistadorul Hernan Cortez a ajuns la marea piramidă aztecă din Cholula. Aici îşi ţineau aztecii o parte din aur. Această piramidă era construită în trepte de o anumită înălţime, având baza un pătrat cu latura de lungime L. Prima treaptă (cea de la baza piramidei) are înălţimea egală cu N, iar celelalte trepte au înălţimea mai mare cu o unitate faţă de cea anterioară. De exemplu, pentru L=7 şi N=4, piramida văzută de sus, respectiv de la sol are următoarea imagine:

Vedere de sus

4	4	4	4	4	4	4
4	5	5	5	5	5	4
4	5	6	6	6	5	4
4	5	6	7	6	5	4
4	5	6	6	6	5	4
4	5	5	5	5	5	4
4	4	4	4	4	4	4

vârful piramidei

baza piramidei= prima treaptă

Vedere de la sol

```
      7
    6 6 6
  5 5 5 5 5
4 4 4 4 4 4 4
```

În timpul luptelor, Cortez prinde un aztec care deconspiră informaţii despre comoara din piramidă. Pentru a afla cantitatea de aur, Cortez trebuie:

- să calculeze suma valorilor de pe fiecare linie a tabloului, reprezentând piramida văzută de sus; Cortez calculează:

4+4+4+4+4+4+4=28
4+5+5+5+5+5+4=33
4+5+6+6+6+5+4=36
4+5+6+7+6+5+4=37
4+5+6+6+6+5+4=36
4+5+5+5+5+5+4=33
4+4+4+4+4+4+4=28

- să lipească toate numerele obţinute anterior (cele 7 sume), pentru a forma cel mai mic număr posibil; Cortez a obţinut 28283333363637.
- din acest număr, să caute cel mai mare număr de două cifre alăturate, acesta reprezentând cantitatea de aur din piramidă. Cortez a calculat şi a aflat: 83!

Scrieţi un program care citeşte numerele naturale nenule N şi L şi care determină:

- numărul obţinut din sume
- cantitatea de aur

Fişierul de intrare piramida.in conţine pe prima linie numerele naturale N şi L, separate printr-un singur spaţiu.

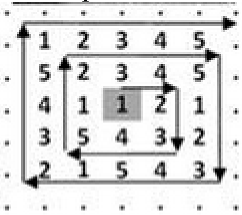
Fişierul de ieşire piramida.out va conţine pe prima linie, numărul cel mai mic obţinut din lipirea sumelor obţinute, iar pe cea de-a doua linie cantitatea de aur.

- $3 \le L \le 9$; $1 \le N \le 50$ şi L număr impar
- pentru rezolvarea primei cerinţe se acordă 60% din punctaj, iar pentru a doua cerinţă 40% din punctaj.

piramida.in	piramida.out
4 5	2020232324 32

79. ***

Rapunzel, frumoasa prinţesă cu păr blond, lung şi magic, se pregăteşte pentru nunta sa cu legendarul Flynn Rider. Cu ocazia acestui eveniment, el îi va dărui prinţesei un medalion unic pe care aceasta îl va purta în ziua nunţii.
Comanda pentru confecţionarea medalionului este preluată de un renumit bijutier al regatului care primeşte de la Flynn k cutii (numerotate de la 1 la k), fiecare cutie conţinând foarte multe cristale, identice ca valoare. Astfel, toate cristalele din prima cutie au valoarea 1, toate cristalele din cea de-a doua cutie au valoarea 2 şi aşa mai departe, astfel încât toate cristalele din ultima cutie au valoarea k. Bijutierul va monta cristalele pe o plachetă de aur de formă pătratică cu n rânduri de cristale, pe fiecare rând fiind montate n cristale unul lângă altul. Acesta ia pe rând câte un cristal din fiecare cutie, în ordinea: 1,2,3,…,k,1,2,3,…,k,1,2,3… şi le aşează pe placheta de aur în formă de spirală. Spirala porneşte din centrul medalionului unde se montează primul cristal. Al doilea cristal semontează în dreapta primului, iar următorul cristal, mai jos, pe rândul imediat următor. Montarea cristalelor se continuă pe acelaşi rând, mergând spre stânga, apoi în sus până la rândul situat deasupra rândului pe care se montează primul cristal. Se procedează în continuare la fel, respectând regula de construire a spiralei: dreapta, jos, stânga, sus şi aşa mai departe. De exemplu, pentru k=5, cristalele se montează pe medalion ca în desenul alăturat.



a) Ştiind că bijutierul completează n (n număr natural impar) rânduri cu cristale pe placheta de aur, determinaţi cea mai mare sumă a valorilor tuturor cristalelor situate pe un rând al medalionului.
b) Determinaţi valoarea cristalului montat cu exact p rânduri deasupra primului cristal montat în medalion şi plasat pe aceeaşi coloană cu acesta.
Fişierul medalion.in conţine pe prima linie numerele naturale k, n, p (în această ordine), cu semnificaţiile menţionate anterior. Valorile k, n, p sunt separate prin câte un spaţiu.
Fişierul medalion.out conţine pe prima linie un număr natural, ce reprezintă numărul determinat conform cerinţei a), iar pe cea de-a doua linie un număr natural, determinat conform cerinţei b).

- 1 < k <= 20
- 2 < n <= 301
- 0 < p <= 500 000
- Numărul de cristale din fiecare cutie este suficient de mare pentru construirea medalionului.

medalion.in	medalion.out
5 3 4	12 2

Explicaţii

a) După montarea a 3 rânduri de cristale se obţine configuraţia:

Sumele valorilor cristalelor de pe aceste rânduri sunt:
2+3+4=9
1+1+2=4
5+4+3=12
Suma cea mai mare este 12.
b) Cristalul situat cu 4 rânduri deasupra centrului medalionului, marcat în desenul alăturat, are valoarea 2.

3	4	5	1	$\boxed{2}$	3	4	5	1
2	3	4	5	1	2	3	4	5
1	2	1	2	3	4	5	1	1
5	1	5	2	3	4	5	2	2
4	5	4	1	1	2	1	3	3
3	4	3	5	4	3	2	4	4
2	3	2	1	5	4	3	5	5
1	2	1	5	4	3	2	1	1
5	4	3	2	1	5	4	3	2