1

Írjatok **C/C++** programot, amely beolvas a billentyűzetről egy **n** (**2<n<20**) természetes számot, majd felépít a memóriában és kiír egy **n** sorból és **n** oszlopból álló mátrixot, melynek sorait és oszlopait **1**-től **n**-ig sorszámozzuk, és amelyben a szegélyen lévő elemek értéke egyenlő az elem sorszámának és oszlopszámának összegével, az összes többi elem egyenlő az előző sorban lévő három „szomszédjának" összegével. Két mátrixbeli elem szomszédos, ha egymás mellett vannak egy sorban, oszlopban, vagy átlón. Egy mátrix szegélyén az első és utolsó sort és oszlopot értjük.

A mátrix elemeit írjuk ki a képernyőre, a mátrix minden sorát a képernyő külön sorába, minden sorban az elemeket szóközzel elválasztva egymástól.

Például: ha **n=5**, akkor a kiírt mátrix : **(10p.)**

```
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
```

2

Írjatok **C/C++** programot, amely beolvas a billentyűzetről egy **n** (**2<n<20**) természetes számot, majd felépít a memóriában és kiír egy **n** sorból és **n** oszlopból álló mátrixot, melynek sorait és oszlopait **1**-től **n**-ig sorszámozzuk, és amelyben minden páratlan sorban lévő elem egyenlő az illető elem sor- és oszlopszámának összegével, és minden páros sorban lévő elem egyenlő az előző sor és ugyanaz az oszlopban lévő elem, és az előző sor és valamelyik szomszédos oszlopban lévő elem közül a legkisebbel.

A mátrix elemeit írjuk ki a képernyőre, a mátrix minden sorát a képernyő külön sorába, minden sorban az elemeket szóközzel elválasztva egymástól.

Például: ha **n=5**, akkor a kiírt mátrix : **(10p.)**

```
2 3 4 5 6
2 2 3 4 5
4 5 6 7 8
4 4 5 6 7
6 7 8 9 10
```

3

Írjatok **C/C++** programot, amely beolvas a billentyűzetről egy **n** (**2<n<20**) természetes számot, majd felépít a memóriában és kiír egy **n** sorból és **n** oszlopból álló mátrixot, melynek sorait és oszlopait **1**-től **n**-ig sorszámozzuk. A mátrix minden páratlan sorában lévő eleme egyenlő lesz a sor sorszámával, és minden páros sorában lévő eleme egyenlő lesz az elem oszlopának sorszámával.

A mátrix elemeit írjuk ki a képernyőre, a mátrix minden sorát a képernyő külön sorába, minden sorban az elemeket szóközzel elválasztva egymástól.

Például: ha **n=5**, akkor a kiírt mátrix : **(10p.)**

```
1 1 1 1 1
1 2 3 4 5
3 3 3 3 3
1 2 3 4 5
5 5 5 5 5
```

4

Írjatok **C/C++** programot, amely beolvas a billentyűzetről egy **n** (**2<n<16**) természetes számot, majd felépít a memóriában és kiír egy **n** sorból és **n** oszlopból álló mátrixot, amelyben a két átlón nullák vannak, a mindkét átló feletti elemek **1**-sek, a mindkét átló alatti elemek **2**–vel egyenlők, az összes többi elem értéke pedig **3**.

A mátrix elemeit írjuk ki a képernyőre, a mátrix minden sorát a képernyő külön sorába, minden sorban az elemeket szóközzel elválasztva egymástól.

Például: ha **n=5**, akkor a kiírt mátrix : **(10p.)**

```
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
```

5

Írjatok **C/C++** programot, amely beolvas a billentyűzetről egy **n** (**2<n<25**) természetes számot, majd feltölt a memóriában egy **n** sorból és **n** oszlopból álló mátrixot, melynek elemeit a következőképpen adjuk meg:

- a mellékátlón az elemek mind nullák lesznek.
- az **i**-dik (**1≤i≤n**) oszlop elemei, a mellékátló felett az **i** értékét veszik fel;
- az **n−i+1**-dik (**1≤i≤n**) sor elemei, a mellékátló alatt az **i** értékét veszik fel.

A program írja ki a mátrixot a képernyőre, a mátrix minden sorát a képernyő külön sorába, egy sor elemeit szóközzel elválasztva egymástól.

Például: ha **n=4** a képernyőn megjelenő mátrix: **(10p.)**

```
1 2 3 0
1 2 0 3
1 0 2 2
0 1 1 1
```

6

Írjatok **C/C++** programot, amely beolvas a billentyűzetről egy **n** (**2<n<25**) természetes számot, majd feltölt a memóriában egy **n** sorból és **n** oszlopból (**1**-től **n**-ig sorszámozva) álló mátrixot, melynek elemeit a következőképpen adjuk meg: a **i.** sorban és **j.** oszlopban található elem értéke a **i*j** (**1≤i≤n** şi **1≤j≤n**) szorzat utolsó számjegye.

A program írja ki az így felépített mátrixot a képernyőre, a mátrix minden sorát a képernyő külön sorába, egy sor elemeit szóközzel elválasztva egymástól.

Például: ha **n=4** a képernyőn megjelenő mátrix: **(10p.)**

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

7

Írjatok **C/C++** programot, amely beolvas a billentyűzetről egy **n** (**2≤n≤24**) természetes számot, majd feltölt a memóriában egy **n** sorból és **n** oszlopból álló mátrixot, melynek elemeit a következőképpen adjuk meg:

- a mellékátlón lévő elemek értéke **0**
- az első sorban lévő elemek értéke, kivéve a mellékátlón találhatót, **n** lesz
- a második sorban lévő elemek értéke, kivéve a mellékátlón találhatót, **n−1** lesz

. . .

– az utolsó sorban lévő elemek értéke, kivéve a mellékátlón találhatót, **1** lesz

A program írja ki a mátrixot a képernyőre, a mátrix minden sorát a képernyő külön sorába, egy sor elemeit szóközzel elválasztva egymástól (lásd a példát).

Például: ha **n=4** a képernyőn megjelenő mátrix: **(10p.)**

```
4 4 4 0
3 3 0 3
2 0 2 2
0 1 1 1
```

8

Írjatok **C/C++** programot, amely beolvas a billentyűzetről egy **n** (**2≤n≤24**) természetes számot, majd feltölt a memóriában egy **n** sorból és **n** oszlopból álló mátrixot, melynek elemeit a következőképpen adjuk meg:

- a főátlón lévő elemek értéke **0**
- az első oszlopban lévő elemek értéke, kivéve a főátlón találhatót, **n** lesz
- a második oszlopban lévő elemek értéke, kivéve a főátlón találhatót, **n−1** lesz

. . .

– az utolsó oszlopban lévő elemek értéke, kivéve a főátlón találhatót, **1** lesz

A program írja ki a mátrixot a képernyőre, a mátrix minden sorát a képernyő külön sorába, egy sor elemeit szóközzel elválasztva egymástól (lásd a példát).

Például: ha **n=4** a képernyőn megjelenő mátrix: **(10p.)**

```
0 3 2 1
4 0 2 1
4 3 0 1
4 3 2 0
```

9

Legyen egy **n** soros és **n** oszlopos (**1≤n≤30**) négyzetes mátrix, mely elemei legtöbb **kétjegyű**, nem nulla egész számok. Írjatok egy **C/C++** programot, mely beolvassa a billentyűzetről **n** értékét, illetve a mátrix elemeit és kiírja a képernyőre a tömb azon elemei szorzatának utolsó számjegyét, amelyek a mellékátlón találhatók és a saját oszlopukban a legkisebbek.

Példa: **n=4**-re és a mellékelt mátrixra a program a képernyőre **1**-et fog kiírni (**3*7=21**). **(10p.)**

3	4	90	10
25	2	7	9
18	3	10	4
3	7	20	3

10

Legyen egy **n** soros és **m** oszlopos (**1≤n≤30**, **1≤m≤30**) mátrix, mely legtöbb **négyjegyű** egész számokat tárol. Írjatok egy **C/C++** programot, mely beolvassa a billentyűzetről **n**, **m** értékeit, illetve a mátrix elemeit, majd kiíratja a képernyőre szóközzel elválasztva minden egyes oszlop legkisebb elemét, az utolsó oszloptól az elsőig haladva.

Példa: **n=4**, **m=4**-re és a mellékelt mátrixra a program a képernyőre kiírja a következő értékeket: **3 7 2 3**. **(10p.)**

3	4	90	10
25	2	7	9
18	3	10	4
3	7	20	3

11

Írjatok egy **C/C++** programot, amely a billentyűzetről beolvas egy **n** (**n<=50**) természetes számot és felépít a memóriában egy **n** soros és **n** oszlopos mátrixot, mely elemei a billentyűzetről beolvasott egész számok. A program kiíratja a képernyőre, a sorrend betartásával, a mátrix minden oszlopának legkisebb elemét. A kiíratott számokat szóközzel választja el egymástól. **(10p.)**

Példa: **n=4**-re és a mellékelt mátrixra, a képernyőre kiírja a következő értékeket: **-7 18 -10 2**.

122	103	5	10
-7	18	-10	2
107	999	59	4
1	200	100	7

12

Írjatok egy **C/C++** programot, amely a billentyűzetről beolvas egy **n** (**n<=20**) természetes számot és felépít a memóriában egy **n** soros és **n** oszlopos mátrixot, mely az első n^2 nullától különböző páros, **3**-al nem osztható természetes számot tárolja. A mátrixot az értékek szerinti növekvő sorrendben töltjük fel, oszloponként és soronként.

A program a mátrix minden sorát a képernyő más sorába írassa ki, az ugyanazon soron lévő elemeket egy-egy szóközzel elválasztva.

Példa: **n=4**-re felépíti és kiíratja a mellékelt mátrixot. **(10p.)**

2	4	8	10
14	16	20	22
26	28	32	34
38	40	44	46

13

Írjatok egy **C/C++** programot, amely a billentyűzetről beolvas egy **n** (**1<=n<=6**) természetes számot és egy **n** soros és **n** oszlopos **A** kétdimenziós tömb elemeit, mely értékei **100**-nál kisebb nullától különböző természetes számok. Írassátok ki a képernyőre az A tömb **"strázsa"** számainak szorzatát. Ha nincs ilyen elem, akkor a **NU EXISTA** üzenetet kell kiíratni.

Egy **x** természetes szám strázsa az **A** tömbben, amennyiben ha beszorozzuk **x**-el az első oszlopban található elemeket, megkapjuk ugyanebben a sorrendben, egy másik oszlop elemeit.

Példa: a mellékelt kétdimenziós tömbre a kiírt eredmény **8**. **(10p.)**

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

14

Írjatok egy **C/C++** programot, amely a billentyűzetről beolvas egy **n** (**2<=n<=9**) természetes számot és egy **n** soros és **n** oszlopos **A** kétdimenziós tömb elemeit, mely értékei **10**–nél kisebb természetes számok. Írassátok ki a képernyőre szóközzel elválasztva a tömb azon elemeit, melyek egyenlők az oszlopában lévő többi elem szorzatával. Ha nem létezik ilyen elem, akkor a **NU EXISTA** üzenetet írja ki.

Példa: a mellékelt tömbre, nem feltétlenül ebben a sorrendben, kiírja a 4 9 (4=1*2*2; 9=3*1*3) értékeket.

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

15

Adott egy **m** sorból és **n** oszlopból álló kétdimenziós tömb (**1≤m≤100, 1≤n≤100**), amelynek elemei a {**0,1,2**} halmazból vannak. Írjatok **C/C++** programot, amely beolvassa a billentyűzetről az **m**, **n** értékeket és a tömb elemeit, majd kiírja azon oszlopok sorszámát, amelyben az elemek szorzata maximális. A tömb sorait **1**-től **m**-ig, oszlopait pedig **1**-től **n**-ig sorszámozzuk. A számokat a képernyőn szóközzel választjuk el egymástól.

Például: ha **m=4** és **n=4** és a beolvasott tömb, akkor a kiírt értékek:
1 2 **(10p.)**

```
2 1 1 0
1 1 1 1
2 2 2 1
1 2 1 1
```

16

Írjatok egy **C/C++** programot, amely a billentyűzetről beolvas egy **n** (**2<n<10**) természetes számot és felépít a memóriában egy **n** soros és **n** oszlopos kétdimenziós tömböt, melyet soronként fentről lefele, illetve oszloponként balról jobbra végigfutva, megkapjuk az első n^2 nem nulla páros számot szigorúan növekvő sorrendben.

Példa: **n=4**-re a program felépíti és kiírja a mellékelt tömböt. **(10p.)**

```
2  4  6  8
10 12 14 16
18 20 22 24
26 28 30 32
```

17

Írjatok egy **C/C++** programot, amely a billentyűzetről beolvas egy **n** (**1<=n<=23**) természetes számot és felépít a memóriában egy **n** soros és **n** oszlopos mátrixot, a következőképpen:

– a főátlón lévő elemek értéke **2**

– a főátló feletti elemek értéke **1**

– a főátló alatti elemek értéke **3**.

A mátrixot írassuk ki a képernyőre, a mátrix minden sorát a képernyő más sorába, az elemeket egy-egy szóközzel elválasztva.
Példa: **n=4**-re a program felépíti és kiíratja a képernyőre a mellékelt mátrixot: **(10p.)**

```
2 1 1 1
3 2 1 1
3 3 2 1
3 3 3 2
```

18

Írjatok egy **C/C++** programot, amely a billentyűzetről beolvas egy **n** (**1≤n≤23**) természetes számot és felépít a memóriában egy **n** soros és **n** oszlopos mátrixot a következőképpen: az **i** (**1≤i≤n**) soron és **j** (**1≤j≤n**) oszlopon található elem egyenlő **i** és **j** összegével.

A program kiíratja a mátrixot a képernyőre, a mátrix minden sorát a képernyő más sorába, az elemeket egy-egy szóközzel elválasztva.
Példa: **n=4**-re a program kiírja a mellékelt mátrixot: **(10p.)**

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

19

Írj egy **C/C++** programot amely beolvassa a billentyűzetről az **n**, **a** és **b** (**2<n<25**, **0<a<n**, **0<b<n**) természetes számokat és felépít a memóriában egy **n** soros és **n** oszlopos mátrixot, (a sorokat és az oszlopokat **1**-től **n**-ig sorszámozzuk). A mátrix csak a **0**, **1**, **2**, **3** és **4** értékeket tartalmazza a következőképpen: az **a** soron vagy **b** oszlopon levő elemek értéke **0**, az **a** sor fölötti, a **b** oszloptól balra levő elemek értéke **1**, az **a** sor fölötti, a **b** oszloptól jobbra levő elemek értéke **2**, az **a** sor alatti, a **b** oszloptól balra levő elemek értéke **3**, az **a** sor alatti, a **b** oszloptól jobbra levő elemek értéke **4**, mint az alábbi példában.

A program írja a képernyőre az így feltöltött mátrixot, minden sort új sorba és egy sor elemeit egy-egy szóközzel elválasztva.
Példa: ha **n=5**, **a=4**, **b=3** akkor a feltöltött és a képernyőre írt mátrix:
(10p.)

```
1 1 0 2 2
1 1 0 2 2
1 1 0 2 2
0 0 0 0 0
3 3 0 4 4
```

20

Írj egy **C/C++** programot, amely beolvassa a billentyűzetről az **n** és **k** (**2<n<25**, **0<k<n**) természetes számokat és létrehozza a memóriában azt az **n** soros és **n** oszlopos mátrixot, amely csak az **1**, **2**, **3** és **4** értékeket tartalmazza a következőképpen: az első **k** sornak és az első **k** oszlopnak a metszéspontjában levő elemek értéke **1**, az első **k** sornak és az utolsó **n−k** oszlopnak a metszéspontjában levő elemek értéke **2**, az utolsó **n−k** sor és az első **k** oszlop metszéspontjában levő elemek értéke **3**, az utolsó **n−k** sor és az utolsó **n−k** oszlop metszéspontjában levő elemek értéke **4**, mint az alábbi példában.

A program írja ki a képernyőre a kapott mátrixot, minden sort külön sorba, az egy sorban levő elemeket egy szóközzel elválasztva.
Példa: ha **n=5**, **k=3** akkor a mellékelt mátrixot kell létrehozni és a képernyőre íratni.
(10p.)

```
1 1 1 2 2
1 1 1 2 2
1 1 1 2 2
3 3 3 4 4
3 3 3 4 4
```

21

Írj egy **C/C++** programot, amely beolvassa a billentyűzetről az **n** (**4≤n≤10**) **páros** természetes számot és egy pontosan **3** számjegyet tartalmazó **x** természetes számot, létrehozza a memóriában azt az **n** soros és **n** oszlopos kétdimenziós tömböt, amelynek főátlóján az **x** szám első számjegye, a mellékátlóján az **x** szám utolsó számjegye, a többi elem helyén az **x** szám középső számjegye van. A kétdimenziós tömböt a képernyőre kell íratni, minden sorát külön sorba, az elemeket szóközzel elválasztva.
Példa: Ha a billentyűzetről beolvasott értékek: **n=4** és **x=123** akkor a mellékelt kétdimenziós tömböt kell kiíratni.
(10p.)

```
1 2 2 3
2 1 3 2
2 3 1 2
3 2 2 1
```

22

Egy **n** soros és **n** oszlopos **a** mátrixban (a sorokat és oszlopokat **1**-től **n**-ig sorszámozzuk) „**k** indexű mellékátlónak” nevezzük a mátrix olyan $a_{i,j}$ elemek halmazát, amelyek esetén a **i+j = k+1** kifejezés igaz, (**1≤k<2*n**, **1≤i≤n**, **1≤j≤n**).
Írj egy **C/C++** programot, amely beolvas a billentyűzetről egy **n** (**2≤n≤24**) természetes számot majd feltölt a memóriában és a képernyőre ír egy **n** soros és **n** oszlopos **a** mátrixot, melyben minden **k** indexű mellékátlóban az elemek értéke $(k+1)^2$, mint a példában.

Az így kapott mátrixot a képernyőre kell írni, minden sort külön sorba, az egy soron belüli elemeket egy-egy szóközzel elválasztva.
Példa: ha **n = 4** akkor a mellékelt mátrixot kapjuk.
(10p.)

```
 4  9 16 25
 9 16 25 36
16 25 36 49
25 36 49 64
```

23

A Fibonacci számsor meghatározása: $f_0 = 0$, $f_1 = 1$, $f_n = f_{n-1} + f_{n-2}$, ha $n>1$.

Írj egy **C/C++** programot, amely beolvas a billentyűzetről egy **n** (**2≤n≤24**) természetes számot, majd feltölt a memóriában és a képernyőre ír egy **n** soros és **n** oszlopos **a** mátrixot, melynek elemei természetes számok, minden elem a Fibonacci számsor egy elemének utolsó számjegye, az **1**-es indexű elemtől az n^2 indexű elemig, sorfolytonosan (egy soron belül balról jobbra, a sorokkal fentről lefele haladva), mint a példában.

A mátrixot a képernyőre kell íratni, minden sort külön sorba, az egyes sorok elemeit egy-egy szóközzel elválasztva.

Példa: ha **n = 4** akkor a mellékelt mátrixot kell kiíratni. **(10 p.)**

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

24

Írj egy **C/C++** programot, amely beolvas a billentyűzetről egy **n** (**2≤n≤24**) természetes számot és felépít a memóriában, majd a képernyőre ír egy **n** soros és **n** oszlopos, a mellékátlóra nézve szimmetrikus **a** mátrixot. A mátrix mellékátlóján és a fölötte elhelyezkedő elemek az **1**, és $\frac{n(n+1)}{2}$ közötti természetes számokat tartalmazzák.

A mellékátló fölött minden sor elemei balról jobbra, és minden oszlop elemei fentről le (beleértve a mellékátló elemeit is) növekvő sorrendben vannak, mint a példában.

A mátrixot a képernyőre kell írni, minden sorát külön sorba, az egyes sorokban levő elemeket szóközzel elválasztva.

Példa: ha **n = 4** akkor a mellékelt mátrixot kell kiíratni. **(10p.)**

```
1 3 6 10
2 5 9 6
4 8 5 3
7 4 2 1
```

25

Írj egy **C/C++** programot, amely beolvas a billentyűzetről két természetes számot az **m** és **n** (**1≤m≤24, 1≤n≤24**) változókba, egy **x** (**1≤x≤m**) számot, majd **m*n** darab legfeljebb 5 jegyű természetes számot, az **m** soros és **n** oszlopos kétdimenziós **a** tömb elemeit. A program törölje a mátrix **x.** sorát, módosítsa megfelelően a sorok számát, majd írja a képernyőre az így kapott kétdimenziós tömböt úgy, hogy a mátrix sorai külön sorban legyenek, valamint az egy sorban levő elemek szóközzel legyenek elválasztva. **(10p.)**

Példa: ha **m=3, n=4, x=2** és a beolvasott mátrix:

```
11 21 31 41
51 61 71 81
91 11 21 31
```

Akkor a kiírt mátrix:

```
11 21 31 41
91 11 21 31
```

26

Írd meg azt a **C/C++** programot, amely beolvas a billentyűzetről egy természetes számot az **n** (**1≤n≤50**) változóba, majd **n*n** darab legfeljebb **5** jegyű természetes számot, az **n** soros és **n** oszlopos kétdimenziós **a** tömb elemeit. A program leellenőrzi hogy a főátló alatti elemek mindegyike nulla-e. Ha igen, akkor a képernyőre a „**Este triunghiulară superior**" üzenetet írja, ellenkező esetben a „**Nu este triunghiulară superior**" üzenetet. **(10p.)**

Példa: ha **n=3** akkor a mellékelt mátrix esetén a kiírt üzenet:

Este triunghiulară superior

```
1 2 3
0 5 6
0 0 9
```

27 Írd meg azt **a C/C++** programot, amely beolvas a billentyűzetről két természetes számot az **m** és **n (1≤m≤50, 1≤n≤50)** változókba, majd **m*n** darab legfeljebb **5** jegyű természetes számot, az **m** soros és **n** oszlopos kétdimenziós tömb elemeit. A program számolja ki és írja a képernyőre azon pozitív elemek szorzatának utolsó számjegyét, melyek sorszáma páros és oszlopszáma páratlan. A sorok és oszlopok sorszámozása **1**-től kezdődik. Ha a páros sorszámú sorokban, valamint páratlan sorszámú oszlopokban nincs pozitív szám, akkor a kiírt üzenet **NU EXISTA**. **(10p.)**

Példa: ha **m=4**, **n=4** akkor a mellékelt mátrix esetén a kiírt érték: **5** **(355=5*71)**.

11	-21	31	41
5	-61	71	-81
91	11	21	31
-11	31	-41	0

28 Írj egy **C/C++** programot, amely beolvas két természetes számot az **n** és **m** (**1≤n≤24**, **1≤m≤24**) változókba és felépíti a memóriában azt az **n** soros és **m** oszlopos kétdimenziós tömböt, amely az **1** és **n*m** közötti minden természetes számot tartalmaz, mint a példában. A program írja a képernyőre, **n** sorba, az így kapott táblázatot, az egy sorban levő elemeket szóközzel elválasztva.

Példa: ha **n=4** és **m=5** akkor a képernyőn:

1	2	3	4	5
10	9	8	7	6
11	12	13	14	15
20	19	18	17	16

29 Írj egy **C/C++** programot, amely beolvassa a billentyűzetről az **n** és **m**, (**1≤n≤24**, **1≤m≤24**) természetes számokat, majd felépíti a memóriában azt az **n** soros és **m** oszlopos kétdimenziós tömböt, amely az összes természetes számot tartalmazza **1**-től **n*m**-ig, úgy, mint a példában. A program írja a képernyőn **n** sorba a kapott táblázatot, az egyes sorok elemeit szóközzel elválasztva.

Példa: ha **n=5** és **m=4** akkor a kiírás:

1	6	11	16
2	7	12	17
3	8	13	18
4	9	14	19
5	10	15	20

30 Írj egy **C/C++** programot, amely beolvas a billentyűzetről egy **n (2<n<=100)** természetes számot és feltölti a memóriában azt az **n** soros és **n** oszlopos négyzetes mátrixot, melynek utolsó sorában az **1,2,3,..,n** számok vannak, ebben a sorrendben, a főátló fölötti elemek értéke **0**. Minden más elem értékét úgy kapjuk meg, hogy összeadjuk a közvetlenül alatta levő sorban a vele egy oszlopban vagy vele szomszédos oszlopokban levő elemek értékét. A program írja a képernyőre az így kapott mátrixot **n** sorba, az elemeket szóközzel elválasztva.

Példa:
Ha **n=4**

(10p.)

Akkor a képernyőn:

27	0	0	0
9	18	0	0
3	6	9	0
1	2	3	4

31

Írjon **C/C++** programot, amely beolvas a billentyűzetről egy **n** (**1≤n≤50**) természetes számot, majd egy **n** soros és **n** oszlopos, legfeljebb **9** számjegyű, zérótól különböző természetes számokat tartalmazó, kétdimenziós tömb elemeit. A program írja a képernyőre a tömb azon elemeit, amelyek szigorúan kisebbek a közvetlen szomszédjaiknál (olyan elemek, melyek azonos sorban, de szomszédos oszlopban, vagy azonos oszlopban de szomszédos sorban vannak). A képernyőre írt értékeket szóközzel kell elválasztani.

Példa: **n=4** és a mellékelt tömb beolvasása esetén a kiírt értékek: **2 0** (a **2** közvetlen szomszédjai: **4**, **3**, **6** és **9**, és ezek mindenikénél kisebb, a **0** közvetlen szomszédjai: **6**, **9** és **1** és ezeknél kisebb).

(10p.)

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

32

Adott egy **n** soros és **m** oszlopos (**1≤n≤24**, **1≤m≤24**) legfeljebb kétjegyű egész számokat tartalmazó kétdimenziós tömb. Írjon **C/C++** programot, amely beolvassa a billentyűzetről az **n** és **m** értékeket, majd a tömb elemeit, és megfordítja minden oszlopban az elemek sorrendjét, mint a példában. A program írja a képernyőre, **n** sorba, az így kapott mátrixot, az egy soron levő elemeket egy-egy szóközzel elválasztva. **(10p.)**

Példa: ha a beolvasott értékek **n=4, m=3** és akkor a képernyőre ezt kell kiíratni:

a mátrix:

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

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

33

Írjon **C/C++** programot, amely beolvassa a billentyűzetről az **m** és **n** (**1≤m≤24**, **1≤n≤24**) természetes számokat, majd egy **m** soros és **n** oszlopos, legfeljebb **4** számjegyből álló, különböző egész számokat tartalmazó kétdimenziós tömb elemeit. A program törölje a tömb legkisebb elemét tartalmazó sort és oszlopot a tömbből (a memóriában). Az így kapott **m−1** soros tömböt a képernyőre kell írni, minden mátrixsort külön sorba, az egy sorban levő elemeket egy-egy szóközzel elválasztva. **(10p.)**

Példa: ha beolvassuk az **m=3** és **n=4** értékeket és az alábbi táblázatot:

```
 2  7  1  4
14  6 12  3
 9 22  8  5
```

képernyőre ezt kell íratni:

```
14  6  3
 9 22  5
```

34

Írjon **C/C++** programot, amely beolvas a billentyűzetről egy **n** (**1≤n≤20**) természetes számot, majd egy **n** soros és **n** oszlopos mátrix elemeit a **[−100, 100]** intervallumból, és a képernyőre írja az **m1−m2** különbséget, ahol **m1** a mátrix főátló fölötti szigorúan pozitív elemeinek számtani középarányosa, **m2** pedig a főátló alatti szigorúan pozitív elemek számtani középarányosa, mint a példában.

Példa: **n=4** és a mellékelt mátrix beolvasása esetén a kiírt érték: **0.25** (**m1=2.75**, a főátló fölötti, kerettel jelölt elemekből számolva, **m2=2.5**, a főátló alatti, aláhúzott elemekből számolva).

(10p.)

−1	2	−4	5
0	6	3	1
2	4	2	0
3	−5	1	−3

35

Írjon **C/C++** programot, amely beolvas a billentyűzetről egy **n** (**1≤n≤20**) természetes számot, majd egy **n** soros és **n** oszlopos mátrix elemeit, egész számokat a **[-100,100]** intervallumból. A program írja a képernyőre a főátló fölötti szigorúan pozitív elemek számtani középarányosát, mint a példában. Ha a főátló felett nem található szigorúan pozitív érték, akkor a **NU EXISTA** üzenetet kell kiíratni.

Példa: **n=4** és a mellékelt mátrix beolvasása esetén a kiírt érték: **2.75** (csak a megjelölt értékekkel számolunk).

(10p.)

-1	$\boxed{2}$	-4	$\boxed{5}$
0	6	$\boxed{3}$	$\boxed{1}$
2	4	2	0
3	-5	1	-3

36

Írjon **C/C++** programot, amely beolvas a billentyűzetről egy **n** (**1≤n≤20**) természetes számot, majd egy **n** soros és **n** oszlopos négyzetes mátrix elemeit, egész számokat a **[-100,100]** intervallumból. A program írja a képernyőre a főátló alatti szigorúan pozitív elemek számtani középarányosát, mint a példában. Ha a főátló alatt nem található szigorúan pozitív érték, akkor a **NU EXISTA** üzenetet kell kiíratni.

Példa: ha a beolvasott érték **n=4**, akkor a mellékelt mátrix esetén a kiírt érték: **2.5** (csak a megjelölt értékekkel számolunk).

(10p.)

-1	2	4	5
0	6	3	1
$\boxed{2}$	$\boxed{4}$	2	0
$\boxed{3}$	-5	$\boxed{1}$	-3

37

Írjon **C/C++** programot, amely beolvas a billentyűzetről egy **n** (**1≤n≤10**) természetes számot, majd felépít a memóriában egy **2*n** soros és **2*n** oszlopos mátrixot (a sorokat és oszlopokat **1**-től **2*n**-ig sorszámozzuk) úgy, hogy a mátrixnak csak a páratlan sorait fentről lefele bejárva, minden páratlan sorban, balról jobbra haladva, az összes **[1,4*n²]** intervallumbeli páratlan számot kapjuk, szigorúan növekvő sorrendben, és a mátrixnak csak a páros sorait fentről lefele bejárva, minden páros sorban, jobbról balra haladva, az összes **[1,4*n²]** intervallumbeli páros számot kapjuk szigorúan növekvő sorrendben, mint a példában.

A program írja a képernyőre a kapott mátrixot, minden mátrixsort külön sorba a képernyőn, az egy sorban levő elemeket egy-egy szóközzel elválasztva.

Példa: ha a beolvasott érték **n=2**, akkor a mellékelt mátrixot kapjuk.

(10p.)

1	3	5	7
8	6	4	2
9	11	13	15
16	14	12	10

38

Írjon **C/C++** programot, amely beolvas a billentyűzetről három nem nulla természetes számot a **k**, **n**, **m** változókba (**n≤10**, **m≤10**, **k≤32000**) majd **n*m** darab, legfeljebb **4** számjegyű egész számot, egy kétdimenziós tömb elemeit (a tömb sorait **1**-től **n**-ig, oszlopait **1**-től **m**-ig sorszámozzuk).

A program határozza meg és írja a képernyőre azon oszlopok sorszámának szorzatát, melyek legalább egyszer tartalmazzák a **k** értéket. Ha nem lézezik egyetlen olyan oszlop sem, amely legalább egyszer tartalmazza a **k** értéket, akkor a program írja ki a **NU EXISTA** üzenetet.

Példa: **k=3**, **n=5**, **m=4** és a mellékelt mátrix beolvasása esetén a kiírt érték **6**, mert a 2. és 3. oszlop tartalmazza a **k=3** értéket. **(10p.)**

2	4	5	-8
5	3	7	9
6	-2	3	10
7	4	2	37
8	7	3	13

39 Írjon **C/C++** programot, amely beolvas a billentyűzetről három nem nulla természetes számot a **k**, **n**, **m** változókba (**n≤10**, **m≤10**, **k≤32000**), majd **n*m** darab, legfeljebb **4** számjegyű egész számot, egy kétdimenziós tömb elemeit (a tömb sorait **1**-től **n**-ig, oszlopait **1**-től **m**-ig sorszámozzuk).

A program határozza meg és írja a képernyőre azon oszlopok sorszámának összegét, melyek legalább egyszer tartalmazzák a **k** értéket. Ha nem lézezik egyetlen olyan oszlop sem, amely legalább egyszer tartalmazza a **k** értéket, akkor a program írja ki a **NU EXISTA** üzenetet.

Példa: **k=3, n=5, m=4** és a mellékelt mátrix beolvasása esetén a kiírt érték **5**, mert a 2. és a 3. oszlop tartalmazza a **k=3** értéket. **(10p.)**

```
2   4   5  -8
1   3   7   9
4  -2   3  10
5   4   2  37
6   7   3  13
```