

4. feladat: Kastély (35 pont)

Egy kastélyban nyolcszög alakú termek vannak. A termeket a bal felső saroktól kezdődően sorszámozzuk. A látogatóknak minden terembe belépéskor adott összeget kell fizetniük (a kiinduló helyen nem kell fizetni). A termek között az ábrának megfelelően négyzet alakú rejtekkajtók nyílnak, amelyek használatáért egységesen R forintot kell fizetni.

Készíts programot (KASTELY.PAS, KASTELY.C, ...), amely kiszámítja a legkisebb költséget, amelybe egy kiinduló teremből egy adott másik terembe eljuthatunk!

A KASTELY.BE szöveges állomány első sorában három egész szám van, a sorok és oszlopok száma ($1 \leq N, M \leq 200$), és a rejtekkajtók költsége ($1 \leq R \leq 100000$) értéke. A második sor 4 egész számot tartalmaz (egy-egy szóközzel elválasztva), az indulási hely sor- és oszlopindexét, valamint a célhely sor- és oszlopindexét. A következő N sor mindegyike M egész számot tartalmaz, ahol az i-edik sor j-edik száma az (i,j) indexű terembe belépés költsége (1 és 10000 közötti egész számok).

A KASTELY.KI szöveges állomány első és egyetlen sorába a legkisebb költséget kell írni, amennyiért eljuthatunk a kezdő teremből a célhelyre!

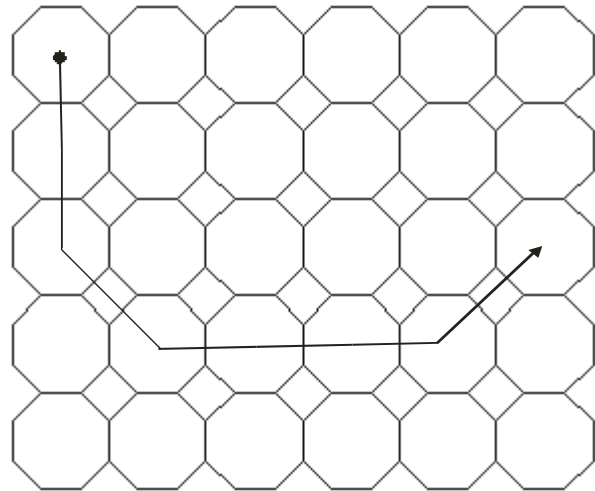
Példa:

KASTELY.BE

```
5 6 40
1 1 3 6
100 100 100 100 100 100
100 100 100 100 900 900
100 900 900 900 900 100
100 100 100 100 100 100
900 900 900 900 900 900
```

KASTELY.KI

780



5. feladat: Közért (30 pont)

L közért K pékségtől rendelhet kenyeret. Megadjuk, hogy az egyes közérték mennyi kenyérré tartanak igényt, és hogy az egyes pékségek mennyit sütnek naponta. Továbbá adott az is, hogy az egyes közérték mely pékségekkel állnak kapcsolatban (csak ilyentől rendelhetnek). A közérték csak egyetlen egy pékségtől rendelhetnek (az adott napon).

Készíts programot (KOZERT.PAS, KOZERT.C, ...), amely megadja, hogy az adott napon melyik közért melyik pékségtől vegye meg a szükséges mennyiségű kenyeret!

A KOZERT.BE szöveges állomány első sorában két egész szám van, a közérték száma ($1 \leq L \leq 30$) és a pékségek száma ($1 \leq K \leq 7$). A második sorban pontosan L pozitív egész szám van, egy-egy szóközzel elválasztva, az egyes közérték igényelt kenyérmennyisége (1 és 1000 közötti egész szám). A harmadik sorban pontosan K pozitív egész szám van, az egyes pékségek napi kenyérsütési kapacitása. A következő L sor egy-egy közért kapcsolatait írja le. Mindegyik sor első száma azon pékségek P_i száma, amelyekkel a közért kapcsolatban van ($1 \leq P_i \leq 5$). Ezt a közérttel kapcsolatban levő P_i pékség sorszáma követi.

A KOZERT.KI szöveges állomány első és egyetlen sorába L egész számot kell írni, egy-egy szóközzel elválasztva, ahol az i -edik szám azon pékség sorszáma legyen, ahonnan az i -edik közért a kenyeret rendeli! Ha több megoldás is van, bármelyik kiírható. Ha nincs megoldás, akkor a sorba egyetlen -1 -et kell kiírni!

Példa:

KOZERT.BE	KOZERT.KI
6 3	3 2 3 1 3 3
50 50 100 300 300 300	
300 1000 750	
2 1 3	
1 2	
1 3	
2 1 3	
2 1 3	
2 1 3	

6. feladat: Beosztás (35 pont)

Egy tehetségkutató versenyre N ember jelentkezett. A zsűri K napon át hallgatja meg a jelentkezőket, naponta legalább egyet, de biztosan a jelentkezések sorrendjében. A zsűrielnök kíváncsi arra, hogy hány ilyen beosztás lehetséges, s ha egy beosztást már ismer, akkor valamilyen szisztematikus módszerrel előállíthatja-e az előzőt, illetve a következőt. Egy beosztás akkor legyen későbbi egy másik beosztásnál, ha a naponkénti létszámai sorozata lexikografikusan kisebb, mint a másik beosztásé.

Készíts programot (BEOSZT.PAS, BEOSZT.C, ...), amely kiszámítja hogy hány ilyen beosztás van, valamint egy adott beosztásra megadja a lexikografikusan következő és előző beosztást!

A BEOSZT.BE szöveges állomány első sorában két egész szám van, a jelentkezők száma ($1 \leq N \leq 50$) és a napok száma ($1 \leq K \leq 10$). A második sor pontosan K egész számot tartalmaz (egy-egy szóközzel elválasztva), az egyes napokon egy beosztásban meghallgatott jelentkezők száma.

A BEOSZT.KI szöveges állomány első sorába a lehetséges beosztások számát kell írni! A második sorba a bemenetről kapott beosztást lexikografikusan megelőző, a harmadik sorba pedig a lexikografikusan követő beosztás kerüljön! Az előzőt és a következőt ciklikusan értjük, azaz az utolsót az első követi, az első az utolsó előzi meg.

Példa:

BEOSZT.BE

5 3
1 3 1

BEOSZT.KI

6
2 1 2
1 2 2

Magyarázat:

1.	2.	3.	4.	5.
1.	2.	3.	4.	5.
1.	2.	3.	4.	5.
1.	2.	3.	4.	5.
1.	2.	3.	4.	5.
1.	2.	3.	4.	5.