

4. feladat: Posta (30 pont)

Nevesincs ország postája csak olyan levél kézbesítését vállalja, amelyen legfeljebb K darab bélyeg van.

Írj programot (**posta.pas**, **posta.c**, **posta.cpp**), amely kiszámítja, hogy melyek azok az I és M közötti értékek, amelyeket nem lehet legfeljebb K bélyeg értékének összegeként előállítani!

Bemenet

A **posta.be** szöveges állomány első sorában három egész szám van, a bélyeg címletek száma ($1 < N \leq 100$), az egy levélre ragasztható bélyegek maximális száma ($1 \leq K \leq 100$), valamint az M érték ($1 \leq M \leq 1000$). A második sor tartalmazza az N darab címlet értékét, minden érték 1 és 1000 közötti egész szám.

Kimenet

A **posta.ki** szöveges állomány első sorába azon I és M közötti értékek R számát kell írni, amelyeket nem lehet legfeljebb K bélyeg értékének összegeként előállítani! A második sor pontosan R egész számot tartalmazzon egy-egy szóközzel elválasztva.

Példa bemenet és kimenet:

posta.be	posta.ki
6 3 50	5
2 3 11 13 17 25	1 10 12 46 48

5. feladat: Szoborpark (30 pont)

Egy szoborparkban N szobor található. A szobrokat mindkét irányban járható utak kötik össze.

Írj programot (**szobor.pas**, **szobor.c**, **szobor.cpp**), amely megadja, hogy a bejáratól Árpád vezér szobráig hány különböző úton juthatunk el!

Bemenet

A **szobor.be** szöveges állomány első sorában négy egész szám van, a csomópontok száma ($1 < N \leq 25$), az utak száma ($0 \leq M \leq 60$), a bejárat csomópont és Árpád vezér szobra csomópont sorszáma ($1 \leq B \neq A \leq N$). A további M sor mindegyike egy $U V$ egész számpárt tartalmaz; ami azt jelenti, hogy az U csomópont és a V csomópont között vezet útszakasz. Teljesül, hogy $1 \leq U \neq V \leq N$.

Kimenet

A **szobor.ki** szöveges állomány egyetlen sorába a kezdőpontból az Árpád vezér szobrához vezető utak számát kell írni!

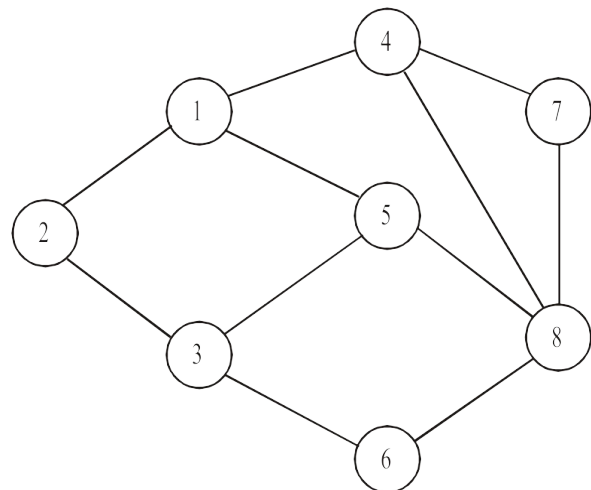
Példa bemenet és kimenet:

szobor.be

```
8 11 2 7
2 3
2 1
1 5
1 4
3 5
3 6
4 7
4 8
5 8
6 8
8 7
```

szobor.ki

13



6. feladat: Kereskedő (40 pont)

Bergengócia M járásában bármely város kereskedője árulhat a járás bármely másik városában. Az uralkodó szeretné elérni, hogy az ország bármely városban elkészített áru bármely városban megvásárolható legyen. Jelentkezett K kereskedő, akik megadták, hogy mely város-párok között mekkora összegért vinnék el az árut.

Írj programot (**keresked.pas**, **keresked.c**, **keresked.cpp**), amely megadja, hogy a király az ajánlatok közül mely kereskedők ajánlatát fogadja el, hogy az a királynak a lehető legkevesebbe kerüljön! A kereskedők egy járáson belül bármely városból bármely városba szállíthatnak árut, aminek nincs további költsége.

Bemenet

A **keresked.be** szöveges állomány első sorában három egész szám van, a járások száma ($1 < M \leq 100$), a települések száma ($M \leq N \leq 10000$), valamint az ajánlatok száma ($1 \leq K \leq 1000$). A következő M sor egy-egy járás településeinek sorszámát tartalmazza, egy-egy szóközzel elválasztva. A további K sor mindegyike egy $U V W$ egész számhármast tartalmaz; ami azt jelenti, hogy az U és a V város között W összegért szállítana a kereskedő árut. Teljesül, hogy $1 \leq U \neq V \leq N$, $0 < W \leq 1000$.

Kimenet

A **keresked.ki** szöveges állomány első sorába a feladat megoldásához szükséges legkisebb költséget kell írni! A második sorba azon ajánlatok sorszámát kell írni, amelyek a legkisebb költség eléréséhez elfogadhatunk. Ha a feladat nem oldható meg, akkor az egyetlen sorba a 0 értéket kell írni!

Példa bemenet és kimenet:

keresked.be

```
3 8 4
1 2 8
4 6
3 5 7
1 4 100
1 6 200
3 8 50
5 2 150
```

keresked.ki

```
150
1 3
```

