

7. feladat: Egérút (50 pont)

Egy irányított gráf egyik pontjából egy macska, másiktól pedig egy egér indul el. Mindkettőre tudjuk, hogy melyik élet mennyi idő alatt teszi meg. Egérlyukak azok a pontok, ahonnan nem vezet tovább él.

Készíts programot (**egerut.pas**, **egerut.c**, ...), amely megadja az egér egy olyan útját, amelyen biztosan beér egy egérlyukba, mielőtt a macska utolérné, bármerre is megy a macska! (Az egérlyuknál sem érheti utol!) A macska elfogja az egeret egy P pontban, ha előbb vagy egy időben ér a P pontba, mint az egér.

Bemenet

Az **egerut.be** szöveges állomány első sorában a pontok ($1 \leq N \leq 100$) és az élek ($1 \leq M \leq 10000$) száma van, egyetlen szóközzel elválasztva. A második sorban a macska és az egér kezdőpontja sorszáma van ($1 \leq A \neq B \leq N$). A következő M sor mindegyikében egy él kezdő- és végpontja sorszáma ($1 \leq K \neq V \leq N$), valamint az az idő van, ami alatt a macska, illetve az egér végighaladhat az élen ($1 \leq IM, IE \leq 100$), egy-egy szóközzel elválasztva.

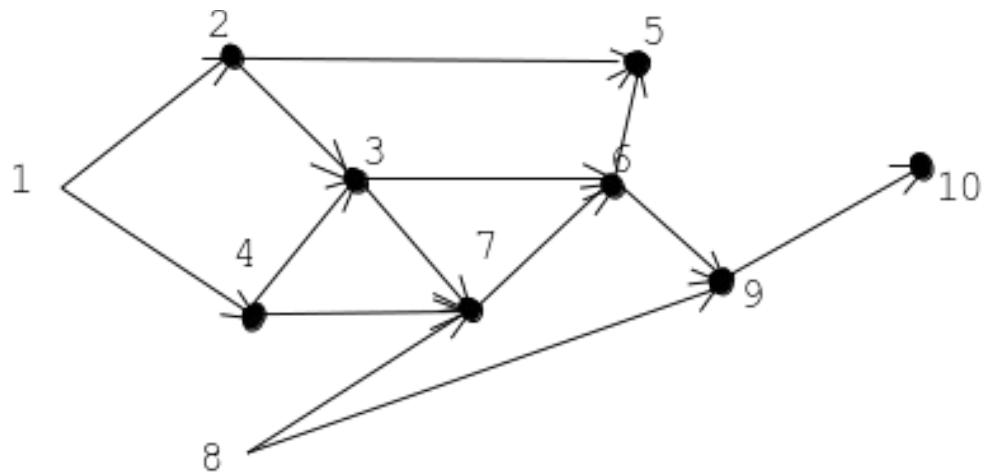
Kimenet

Az **egerut.ki** szöveges állományba egy olyan út sorszámait kell írni, amelyen az egér biztonságosan beér egy egérlyukba! Ha nincs ilyen, akkor az első sorba az egyetlen 0 számot kell írni, ha több megoldás is lenne, akkor bármelyik kiírható.

Példa bemenet és kimenet:

```
egerut.be
10 14
1 8
1 2 1 2
1 4 1 2
2 3 1 2
4 3 1 2
2 5 1 2
3 6 1 2
3 7 1 2
4 7 1 2
6 9 1 2
7 6 1 2
8 7 1 2
8 9 1 2
9 10 1 2
6 5 1 2
```

```
egerut.ki
8 9 10
```



8. feladat: Csomag (50 pont)

A csomagküldő szolgálat központjában a beérkezés sorrendjében várakoznak a csomagok továbbításra. Minden csomagnak ismert a súlya, ezek a beérkezés sorrendjében: s_1, \dots, s_N . A cégnek két kamionja van, mindegyik azonos K kapacitású, tehát mindegyikre legfeljebb annyi csomag pakolható, hogy a csomagok összsúlya nem lehet K -nál nagyobb. Minden csomag súlya legfeljebb K . A lehető legtöbb csomagot akarják továbbítani a két kamionnal. Tehát kiszámítandó az a legnagyobb M ($1 \leq M \leq N$), hogy a sorban első M csomag mindegyike felpakolható a két kamion valamelyikére. Az ilyen pakolást nevezzük optimálisnak.

Írj programot (**csomag.pas**, **csomag.c**, ...), amely kiszámít egy optimális pakolást!

Bemenet

Az **csomag.be** szöveges állomány első sora két egész számot tartalmaz (egy szóközzel elválasztva), a kamion K ($1 \leq K \leq 200$) kapacitását és a csomagok N számát ($1 \leq N \leq 200$). A második sor pontosan N egész számot tartalmaz (egy-egy szóközzel elválasztva). A sorban i -edik szám az i -edik csomag súlya. Minden s csomag súlyra teljesül, hogy $1 \leq s \leq K$.

Kimenet

A **csomag.ki** szöveges állomány első sora azt a legnagyobb M ($1 \leq M \leq N$), indexet tartalmazza, amelyre teljesül, hogy az első M csomag felpakolható a két kamionra, betartva a K súlykorlátot! A második és harmadik sor azoknak a csomagoknak a sorszámaikat tartalmazza, amelyeket az első, illetve a második kamionra pakolnak egy optimális pakolás során.

Példa bemenet és kimenet:

csomag.be	csomag.ki
100 6	3
12 45 64 56 23 42	1 3
	2