1. feladat: Pakolás (30 pont)

Egy raktárban üres ládák állnak egy sorban. ***N*** különböző méretű láda van, minden láda kocka alakú és a felső oldala nyitott. A ládákat össze akarják egymásba pakolni. Az összepakolás az alábbi szabályok szerint történhet. Az ***i***-edik helyen lévő ládákat berakhatjuk a ***j***-edik helyen lévő ládákba, ha az ***i***-edik és ***j***-edik hely között már nincs láda, és a ***j***-edik helyen lévő (esetleg már összepakolt) ládákba belefér, tehát az ***i***-edik helyen lévő ládák legnagyobbikának mérete kisebb, mint a ***j***-edik helyen összepakolt ládák legkisebbike.

Írj programot (**`pakol.pas`**, **`pakol.c`**, **`pakol.cpp`**), amely kiszámítja, hogy összerakhatók-e a ládák egybe, és megad egy összepakolási műveletsort!

Bemenet

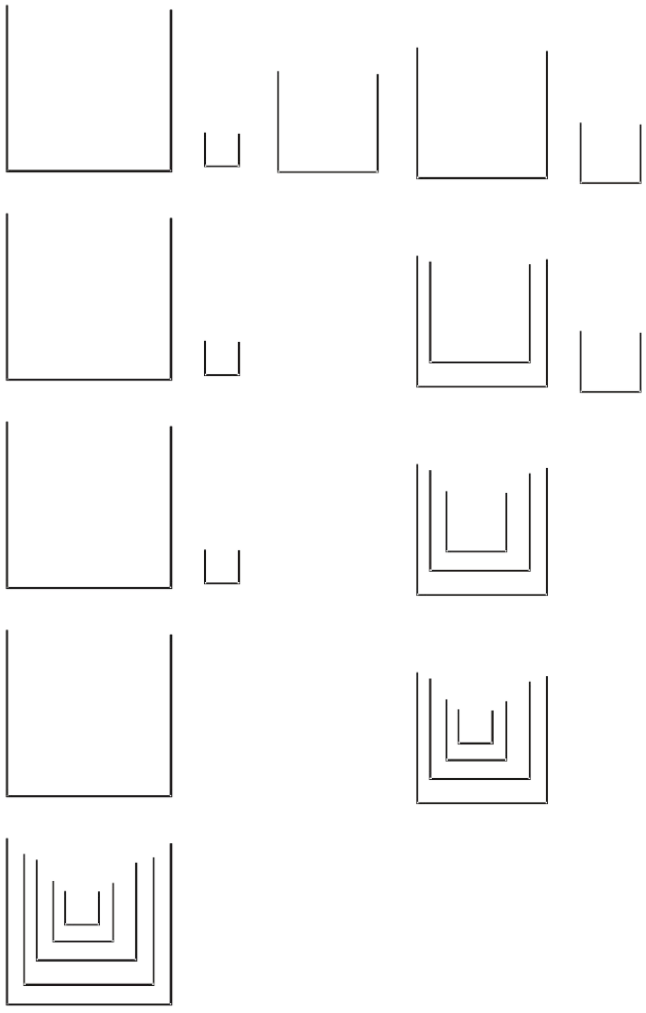
A **`pakol.be`** szöveges állomány első sorában egy egész szám van, a ládák száma (***1≤N≤10000***). A második sor pontosan ***N*** különböző pozitív egész számot tartalmaz, a ládák méretét. Az ***i***-edik szám az ***i***-edik láda mérete, ami ***1*** és ***N*** közötti egész szám.

Kimenet

A **`pakol.ki`** szöveges állomány első sorába két ***0*** számot (egy szóközzel elválasztva) kell írni, ha nem lehet az összes ládát egybe összepakolni. Egyébként pontosan ***N-1*** sort kell kiírni, soronként egy pakolási művelet két pozícióját, ***u***-t és ***v***-t, ami azt jelenti, hogy az **u** pozíción lévő ládákat a ***v*** pozíción lévő ládákba kell belerakni.

Példa bemenet és kimenet:

pakol.be	pakol.ki
5	3 4
5 1 3 4 2	5 4
	2 4
	4 1



2. feladat: Verseny (30 pont)

Egy kerékpárversenyen a versenyzők egy labirintusszerű versenypályán haladnak. A labirintus csomópontjai közötti egyes útszakaszok csak egy irányban járhatóak, s tudjuk, hogy bármilyen útszakaszt is választanak, biztosan célba érnek. A kezdőpont az a csomópont, ahova nem vezet útszakasz. A cél az a csomópont, ahonnan nem vezet ki útszakasz.

Írj programot (**verseny.pas**, **verseny.c**, **verseny.cpp**), amely megadja, hogy hány különböző úton lehet eljutni a célba!

Bemenet

A **verseny.be** szöveges állomány első sorában négy egész szám van, a csomópontok száma (***1<N≤100***), az utak száma (***0≤M≤1000***), a start csomópont és a cél csomópont sorszáma ***(1≤S≠C≤N)***. A további M sor mindegyike egy ***U V*** egész számpárt tartalmaz; ami azt jelenti, hogy az ***U*** csomópontból a ***V*** csomópontba vezet útszakasz. Teljesül, hogy ***1≤U≠V≤N.***

Kimenet

A **verseny.ki** szöveges állomány egyetlen sorába a kezdőpontból a célpontba menő utak számát kell írni!

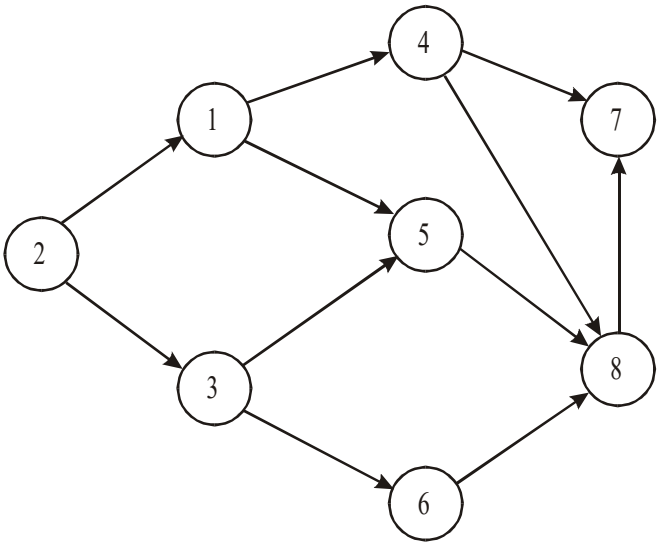
Példa bemenet és kimenet:

verseny.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
```

verseny.ki

```
5
```



3. feladat: ÁdámÉva (40 pont)

Ádám és Éva két különböző városban élnek. A városok között egyirányú utakon lehet közlekedni. Ádám és Éva találkozni szeretnének valahol. Mivel Ádám udvarias, úgy döntött, hogy olyan városban kellene találkozniuk, ahova Évának a lehető legkevesebbet kell utaznia. Ha több ilyen város is lenne, akkor Ádám már magára is gondolhat, azaz ezek közül azt kell választani, ahova Ádám a lehető legkevesebb utazással eljuthat.

Írj programot (**adameva.pas, adameva.c, adameva.cpp**), amely megadja, hogy Ádám és Éva melyik városban találkozzon!

Bemenet

Az **adameva.be** szöveges állomány első sorában négy egész szám van, a városok száma (***1<N≤100***), az utak száma (***0≤M≤1000***), valamint Ádám és Éva városának sorszáma (***1≤A≠E≤N***) van. A további M sor mindegyike egy ***U V*** egész számpárt tartalmaz; ami azt jelenti, hogy az ***U*** városból a ***V*** városba vezet közvetlen út. Teljesül, hogy ***1≤U≠V≤N.***

Kimenet

Az **adameva.ki** szöveges állomány első sorába három egész számot kell írni. Az első szám annak a ***T*** városnak a sorszámát, ahol Ádám és Éva találkozni fog. A második szám Ádám útvonalának hossza, a harmadik pedig Éva útvonalának hossza legyen. A második sor Ádám útvonalát, a harmadik pedig Éva útvonalát tartalmazza. Több megoldás esetén bármelyik megadható. Ha Ádám és Éva nem találkozhatnak, akkor az állomány egyetlen sorába egy 0-t kell kiírni!

Példa bemenet és kimenet:

adameva.be

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

adameva.ki

```
5 1 1
1 5
2 5
```