

7. feladat: Robot

Egy automatizált üzemben a gyártó berendezések egy nagy gépcsarnokban négyzetrácsos elrendezésben vannak. A műszak végeztével egy robot gyűjti össze a legyártott alkatrészeket, amely a berendezések felett a rácsszerkezet által meghatározott pályán mozoghat, csak jobbra, lefelé vagy felfelé tud mozogni. Egy lépés megtételéhez 1 időegységre (másodperc) van szüksége, egyszerre tetszőleges számú alkatrész tud szállítani. A robot induláskor a rácsszerkezet bal felső sarkában van, és a jobb alsó sarokba kell eljuttatnia a legyártott alkatrészeket.

Írj programot (ROBOT.PAS), amely kiszámítja, hogy minimálisan hány időegység alatt tudja egy robot begyűjteni az összes legyártott alkatrészt!

A feladat megoldásához rendelkezésre áll az üzemcsarnok térképe, amely azt írja le, hogy a rácsszerkezet mely helyein található a gyártó gépek.

A ROBOT.BE állomány első sora a rácsszerkezet méretét megadó M és N értékeket tartalmazza egy szóközzel elválasztva ($1 \leq M, N \leq 1000$). M a rácsszerkezet sorainak, N pedig az oszlopainak a száma. Az állomány ezt követő soraiban a gyártó gépek sor- és oszlopkoordinátái vannak: minden sorban egy $a\ b$ ($1 \leq a \leq M, 1 \leq b \leq N$) számpár, egyetlen szóközzel elválasztva. A bemeneti állományt a $0\ 0$ számpárt tartalmazó sor zárja. A bal felső sarok koordinátái $(1, 1)$, a jobb alsóé pedig (M, N) .

A feladat megoldását adó egyetlen számot a ROBOT.KI állomány első sorába kell írni.

Példa:

ROBOT.BE

```
6 5
1 3
2 1
4 1
3 3
5 3
6 5
0 0
```

ROBOT.KI

15

8. feladat: Tartozások

Egy ország vállalatai közül sokan tartoznak másoknak, de a tartozást nem tudják megadni, mert mások is tartoznak nekik.

Előfordulhat például olyan eset, hogy A tartozik B-nek 1 millió forinttal, B C-nek szintén 1 millióval, s C A-nak 2 millióval. Ebben az esetben azonban elég, ha C fizet A-nak 1 millió forintot, s a maradék 1 milliós körbetartozást (mivel mindenkinek ugyanannyi a követelése és a tartozása egymással szemben) el lehet felejtetni.

Készíts programot (TARTOZ.PAS), amely a vállalatok tartozásait a lehető legjobban egyszerűsíti: a tartozások száma a lehető legkisebb legyen, s ezen belül a tartozási összegek is a lehető legkisebbé váljanak!

A TARTOZ.BE állomány első sorában a vállalatok száma ($1 \leq N \leq 100$) van, további soraiban (legfeljebb 1000) pedig a tartozások adatai szerepelnek: ki kinek mennyivel tartozik ($1 \leq ki \leq N$, $1 \leq kinek \leq N$, $0 < mennyivel \leq 10000000$), egy-egy szóközzel elválasztva. (Ugyanaz a vállalatpár többször is szerepelhet.)

A TARTOZ.KI állomány soraiba ugyanilyen formátumban kell írni az egyszerűsített tartozásokat!

Példa:

TARTOZ.BE	TARTOZ.KI
3	1 3 40
1 2 20	2 3 10
1 3 40	
2 3 20	
1 2 10	
3 1 10	
2 1 20	

9. feladat: Elfogó

A városi rendőrség egy körözött bűnöző elfogását tervezi, aki autóval folyamatosan közlekedik a város utcáin. Minden utca egyirányú, bármely két útkereszteződést egy irányban legfeljebb egy utca köt össze, továbbá a városban lehet zsákutca is, azaz olyan útkereszteződés, amelyből nem vezet kifelé egyirányú utca.

Ha van olyan útkereszteződés, amelyben a bűnözőnek előbb-utóbb fel tűnnie, akkor az elfogásához elegendő a rendőröket egy ilyen helyre összpontosítani.

Írj programot (ELFOGO.PAS), amely meghatározza az összes olyan útkereszteződést, ahol a bűnözőnek előbb-utóbb fel kell bukkannia, bárhonnan induljon is, feltéve, hogy folyamatosan közlekedik.

A ELFOGO.BE állomány első sora az útkereszteződések N számát tartalmazza ($2 \leq N \leq 200$). Az útkereszteződéseket a $1, \dots, N$ természetes számokkal azonosítjuk. Az állomány második sorában az egyirányú utcák száma (M) van. Az állomány ezt követő M sora az egyirányú utcákat adja meg. Az egyes sorokban az A -ból B -be vezető utcát az $A B$ számpár írja le, egyetlen szóközzel elválasztva ($1 \leq A, B \leq N$).

A ELFOGO.KI állomány egyetlen sorába az összes olyan útkereszteződés azonosító számát kell írni (egy-egy szóközzel elválasztva), amelyen a bűnözőnek előbb-utóbb fel kell bukkannia. Ha nincs ilyen útkereszteződés, akkor a sor tartalma a egyetlen 0 szám legyen.

Példa:

ELFOGO.BE	ELFOGO.KI
6	1 5
8	
1 2	
2 3	
3 4	
4 5	
1 6	
2 5	
6 3	
6 4	