

1. feladat: Csoki (30 pont)

Egy N tagú társaság csokoládéüzletben vásárol. Az üzletben egy sorban M dobozban található csokoládé. A céljuk az, hogy egymás melletti dobozokból az összes csokit megvegyék úgy, hogy mindenkinek ugyanannyi csoki jusson!

Készíts programot (CSOKI.PAS, CSOKI.C vagy CSOKI.CPP), amely meghatározza, hogy hányadik doboztól hányadik dobozig kell megvenniük az összes csokit, hogy mindenkinek ugyanannyi csoki jusson!

A CSOKI.BE állomány első sorában a társaság tagjai száma ($1 \leq N \leq 1000$), valamint a dobozok száma ($1 \leq M \leq 100000$) van. A következő M sor mindegyike egy egész számot tartalmaz, egy-egy dobozban levő csokik számát (legfeljebb 10000).

A CSOKI.KI állomány első sorába annak a két doboznak a sorszámát kell írni, amelyek közötti összes csokit megvéve mindenki ugyanannyi csokit kaphat. Ha a feladatnak több megoldása is lenne, akkor közülük egy olyat kell kiírni, amelynél a legkevesebb csokit kell megvenni! Ha nincs megoldás, akkor az egyetlen 0 számot kell kiírni!

Példa:

CSOKI.BE	CSOKI.KI
5 5	2 4
4	
2	
7	
6	
3	

2. feladat: Benzin (30 pont)

Egy $N \cdot M$ -es téglalapnyi méretű sivatagon kell átjutni autóval, a sivatag bal felső (1,1) sarkától a jobb alsó (N,M) sarkáig úgy, hogy csak jobbra és lefelé haladhatunk. Az autó lépésként B liter benzint fogyaszt. A sivatag egyes pontjain benzinkészletet rejtettek el, amit az autó mind magával vihet. A cél: eljutni a jobb alsó sarokba úgy, hogy a lehető legtöbb benzint vigyük oda,

Készíts programot (BENZIN.PAS, BENZIN.C vagy BENZIN.CPP), amely meghatározza, hogy maximum mennyi benzint lehet a kezdőpontból a célpontba eljuttatni!

A BENZIN.BE állomány első sorában a terület sorai és oszlopai száma ($1 \leq N, M \leq 1000$), valamint az egy lépés megtételéhez szükséges benzin mennyisége ($1 \leq B \leq 100$) van, egy-egy szóközzel elválasztva. A következő N sor mindegyike M egész számot tartalmaz, egy-egy szóközzel elválasztva, az i-edik sor j-edik száma az (i,j) pozícióban levő benzin mennyisége ($0 \leq \text{mennyiség} \leq 1000$).

A BENZIN.KI állomány egyetlen sorába A célba eljuttatható benzin maximális mennyiségét kell írni! Ha nincs megoldás, akkor az egyetlen -1 számot kell kiírni! A második sorba pontosan $N+M-2$ karaktert kell kiírni, szóközők nélkül, egy olyan lépéssorozatot, amely a legjobb megoldást adja. Több megoldás esetén bármelyik megadható.

Példa:

BENZIN.BE	BENZIN.KI
6 5 3	12
4 2 3 4 5	JJLJLLJLL
0 2 4 5 2	
3 5 1 0 1	
4 2 0 6 7	
1 2 3 4 5	
0 7 4 2 3	

3. feladat: Raktár (40 pont)

Egy szállítványozási vállalat elosztó raktárában tárolja, majd innen továbbítja a beérkező konténereket. Minden konténerre rá van írva a rendeltetési helye, ami egy sorszámmal van azonosítva. A kiszállítás a rendeltetési helyek szerinti sorrendben történik. A helyszűke miatt a konténereket oszlopokat képezve egymásra kell rakni, de úgy, hogy bármely u sorszámú konténerre nem lehet olyan konténert rakni, amelynek v sorszáma nagyobb, mint u , mert a kiszállításakor így nem kell átpakolni konténereket. Amikor egy konténer beérkezik, akkor azonnal vagy egy már elkezdett oszlop tetejére kell tenni, vagy új oszlopot kell kezdeni, amelynek legelső konténere lesz a beérkezett.

Készíts programot (RAKTAR.PAS, RAKTAR.C vagy RAKTAR.CPP), amely kiszámítja, hogy minimálisan hány oszlopba lehet összerakni a konténereket, és meg is ad egy összerakást!

A RAKTAR.BE állomány első sorában a konténerek N száma ($1 \leq N \leq 100000$), és a lehetséges rendeltetési helyek K ($1 \leq K \leq 10000$) száma van. A második sorban pontosan N egész szám van (egy-egy szóközzel elválasztva), az i -edik szám az i -edik konténer rendeltetési helye.

A RAKTAR.KI állomány első sorába azt a legkisebb M számot kell írni, amelyre teljesül, hogy a konténereket össze lehet rakni M oszlopba! A második sor pontosan N egész számot tartalmazzon (egy-egy szóközzel elválasztva), az i -edik szám annak az oszlopnak a sorszáma legyen, amelyikre az i -edik konténert rakjuk beérkezésekor! Több megoldás esetén bármelyik megadható.

Példa:

RAKTAR.BE

10 9
6 3 4 3 8 1 2 7 3 5

RAKTAR.KI

4
1 1 2 1 3 1 2 3 3 4