

## Előállítások száma

Adott egy rendezett  $N$  elemű sorozat és egy  $K$  egész szám.

Írj programot, amely megadja, hogy a  $K$  hány különböző indexű elempár összegeként állhat elő!

### Bemenet

A *standard bemenet* első sorában a sorozat elemszáma ( $1 \leq N \leq 100\,000$ ) és a kívánt összeg ( $1 \leq K \leq 200\,000$ ) van. A második sor tartalmazza a sorozatot ( $1 \leq S_i \leq S_{i+1} \leq K$ ), időrendben.

### Kimenet

A *standard kimenet* első sorába azt a számot kell írni, ahányféleképpen a  $K$  különböző indexű elempár összegeként előállhat!

### Példa

Bemenet	Kimenet
8 7	6
1 2 2 3 3 4 4 5	

Magyarázat:  $K = x_2 + x_8, x_3 + x_8, x_4 + x_6, x_4 + x_7, x_5 + x_6, x_5 + x_7$ .

### Korlátok

Időlimit: 0.1 mp.

Memórialimit: 32 MB

## Játéktábla

Egy játékban  $N \times M$ -es játéktáblát használnak. Minden játékos a bal felső ((1,1) indexű) sarokból indul, csak jobbra, jobbra lefelé vagy lefelé léphet, a jobb alsó sarokba kell eljutnia. Kezdetben kap  $K$  forint kezdőtőkét a bal felső sarokban, minden lépése  $L$  forintba kerül. Egyes mezőkön pénz található, amit felszedhet. Ha a pénze nem elég a következő lépéshez, nem mehet tovább.

Írj programot, amely megadja a legtöbb pénzt, amivel egy játékos a jobb alsó sarokba érhet!

### Bemenet

A *standard bemenet* első sorában a sorok és oszlopok száma ( $1 \leq N, M \leq 100$ ), valamint egy lépés költsége ( $1 \leq L \leq 10$ ) van. A következő  $N$  sorban levő  $M$  érték az egyes mezőkön levő pénz értékét jelenti ( $1 \leq P_{i,j} \leq 1000$ ).

### Kimenet

A *standard kimenet* első sorába a legtöbb pénzt kell írni, amivel egy játékos a jobb alsó sarokba érhet! Ha nem juthat el a sarokba, akkor -1-et kell írni!

### Példa

Bemenet

```
5 6 2
8 0 0 0 0 9
0 0 0 0 4 2
6 0 0 0 4 0
0 0 0 0 0 5
0 4 0 0 0 0
```

Kimenet

7

Magyarázat: 8 lépést tett meg a piros színű pontokon keresztül, 23 pénzt kapott, 16-ba kerültek a lépések.

### Korlátok

Időlimit: 0.1 mp.

Memórialimit: 32 MB

## Szerencsejáték

Egy szerencsejátékban naponta egyetlen egész számot sorsolnak ki. Ha egy tetszőleges számot nem először sorsolnak ki, akkor megállapítják, hogy hány nap telt el az előző kisorsolása óta. Minden ilyen szakaszra kiszámolják, hogy a sorsolás során hányszor fordult elő.

Írj programot, amely megadja a legtöbbször előfordult hosszúságú szakaszok számát!

### Bemenet

A *standard bemenet* első sorában a napok száma ( $1 \leq N \leq 100\,000$ ) és a sorsolt számok felső határa ( $1 \leq M \leq 1000$ ) van. A következő  $N$  sorban az egyes sorsolások értékei vannak ( $1 \leq S_i \leq M$ ), időrendben.

### Kimenet

A *standard kimenet* első sorába azt kell kiírni, hogy legfeljebb hányszor fordult elő ugyanolyan távolság egy szám előző kisorsolása óta. Ha nincs megoldás, akkor 0-t kell kiírni!

### Példa

Bemenet	Kimenet
8 3	3
1 2 1 3 2 1 3 3	

Magyarázat: Egymástól 3 napra található a második és harmadik 1-es, a két 2-es, valamint az első és második 3-as. Egyetlen 2 távolságos érték van, az első és második 1-es. Egyetlen 1 távolságos érték van, a második és harmadik 3-as. A legtöbb ismétlődő szakasz a 3 hosszúságú, ebből 3 darab van.

### Korlátok

Időlimit: 0.1 mp.

Memórialimit: 32 MB

## Ügyelet

Egy  $N$  napos időszakra  $M$  ember vállalt ügyeletet.

Írj programot, amely megadja a legrövidebb intervallumot, ahol mindenki volt ügyeletes!

### Bemenet

A *standard bemenet* első sorában a napok száma ( $1 \leq N \leq 100\,000$ ) és az emberek száma ( $1 \leq M \leq 1000$ ) van. A következő  $N$  sorban az egyes napokon ügyeletesek sorszámai vannak ( $1 \leq S_i \leq M$ ), időrendben.

### Kimenet

A *standard kimenet* első sorába a legrövidebb intervallum első és utolsó napjának sorszámát kell írni, amely alatt mindenki volt ügyeletes! Több megoldás esetén a legkorábban kezdődőt kell kiírni! Ha nincs megoldás, akkor -1-et kell kiírni!

### Példa

Bemenet	Kimenet
8 3	5 7
1 2 1 1 2 1 3 3	

Magyarázat: Az 5.-től a 7. napig mind a hárman ügyeltek.

### Korlátok

Időlimit: 0.1 mp.

Memórialimit: 32 MB