

1. A hegy térképe egy n soros terület magasságát jelöli. Egy hegymászó az $(1,1)$ koordinátáról indul, és el szeretne jutni az (n,m) koordinátához. Csak sorban vagy oszlopban mozoghat, és egy szomszédos területre csak akkor léphet át, ha annak magassága nem kisebb az aktuális hely magasságánál.



Határozzuk meg a hegymászó maximális útvonalának hosszát!

Bemenet

A program beolvassa a standard bemenetről a n és m számokat, majd egy $n \times m$ természetes számokból álló mátrixot, amely a hegy térképét reprezentálja.

Kimenet

A program kiírja a képernyőre az L értéket, amely a hegymászó maximális útvonalának hosszát jelenti.

Megkötések és megjegyzések

- $1 \leq n, m \leq 201$
- A területek magassága kisebb mint 100;
- A hegymászó nem hagyhatja el a hegyet, és nem léphet kétszer ugyanarra a területre;
- Ha nem létezik érvényes útvonal, akkor az „nincs megoldás” üzenetet kell kiírni.

Példa

Bemenet

```
4 5
4 5 3 8 9
8 7 7 8 9
9 7 3 9 10
2 5 4 10 11
```

Kimenet

10

Magyarázat

A maximális hosszúságú útvonal a következő lehet:

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

Feladat: Adott összeg kiszámítása backtracking algoritmussal

2. Adott egy n elemű természetes számokból álló halmaz, valamint egy S célösszeg. Meg kell határozni, hogy létezik-e olyan részhalmaz, amelynek elemei összeadva pontosan S -t eredményeznek. Ha létezik, akkor adjuk meg egy ilyen részhalmazt!

A program beolvassa az első sorból az n számot és az S célösszeget.

A második sorban n darab természetes szám következik, amelyek a halmaz elemeit alkotják.

Kimenet

- Ha létezik olyan részhalmaz, amelynek összege pontosan S , akkor a program írja ki az egyik ilyen részhalmaz elemeit.
- Ha nincs ilyen részhalmaz, akkor írja ki az „Nincs megoldás” üzenetet.

Példa

Bemenet

```
5 9
2 3 7 1 5
```

Kimenet

```
2 7
3 1 5
```

