

# Lift

A feladat egy lift mozgásának szimulálása. A lift fel-le mozog egy toronyházban, és szállítja az embereket. A lift kapacitása végtelen, viszont mindig az aktuálisan következő kérést igyekszik teljesíteni. A következőt csak akkor, ha azt az előző kérés teljesítésének megzavarása nélkül megteheti (egy szimpla megállás nem zavarás, de egy irányváltás már igen).

## Bemenet

A bemenetet a `feladat.be` fájlból kell olvasni. A bemenet több sorból áll. Az első sorban két szóközzel elválasztott egész érték van,  $M$  és  $N$  ( $1 \leq M \leq 100, 0 \leq N \leq M$ ), amik az emeletek számát és a lift kezdeti pozícióját jelentik. Az emeletek számába a földszint (0. emelet) nincs beleszámolva. A második sor egy egész értéket tartalmaz: a következő sorokban felsorolt kérések számát ( $0 \leq K < 1000$ ). A többi sorban mindenhol két egész érték van ( $0 \leq I, O \leq M$ ) egy-egy szóközzel elválasztva, az első a beszállás, a második pedig a kiszállás helyét jelzi. Ha a beszállás és a kiszállás helye megegyezik, az ajtót akkor is ki kell nyitni!

## Kimenet

A kimenetet a `feladat.ki` fájlba kell írni. A kimenet minden sora pontosan egy  $0$  és  $M$  közötti számot tartalmaz, ami azt jelenti, hogy a lift ezen a szinten kinyitotta az ajtaját. A lift bekapcsolása után a kezdőszinten kinyitja az ajtót. Ezután a lift az alábbi algoritmus szerint működik: (1) Ha van, meghatározza, hogy honnan jön a következő kérés ( $I$ ). (2) Elindul a kérés irányába. Menet közben minden további kérést teljesít (azaz a kérés helyén és a célemeleten is megáll, ajtót nyit), amit irányváltoztatás, és az eredeti kérés emeletén ( $I$ ) való túlfutás nélkül teljesíteni tud. Az első nem teljesíthető kérésnél felfüggeszti a további kérések teljesítését (vagyis ha a sorban van egy nem teljesíthető kérés, az ez utániakkal egyelőre nem foglalkozik, de az előtte lévőköt teljesíti tekintet nélkül a sorrendjükre). (3) Mikor megérkezett az eredeti kérés emeletére meghatározza a célemeletét ( $O$ ). (4) Elindul a kérés irányába. Menet közben sorban minden további kérést teljesít, amit irányváltoztatás nélkül teljesíteni tud az első nem teljesíthető kérésig (vagyis ha a sorban van egy nem teljesíthető kérés, a továbbiakat nem nézi). Ilyenkor ha kell túlfuthat az eredeti kérés célemeletén ( $O$ ), ha egy további kérés miatt ez szükséges. (5) A működés az (1) ponttal folytatódik. Az output minden sorát sorvége jel zárja le.

## Példa 1

### bemenet

```
8 0
6
8 0
8 8
7 7
7 6
```

6 6  
6 5

### **kimenet**

0  
7  
8  
7  
6  
5  
0

## **Példa 2**

### **bemenet**

38 7  
4  
18 9  
20 3  
14 5  
8 37

### **kimenet**

7  
18  
9  
20  
14  
5  
3  
8  
37

## **Példa 3**

### **bemenet**

37 1  
3  
35 2  
36 29

25 30

## **kimenet**

1

35

2

25

30

36

29