

## Lovagi torna

1491-ben Beatrice d'Este és Duke Milan Lodovico Sforza esküvőjén lovagi tornát rendeznek, ami 3 napig tart. A legnépszerűbb lovag még nem érkezett meg.

### A verseny

A verseny kezdetén az  $N$  lovagot felsorakoztatják egy sorba, a pozíciójukat 0-tól  $N-1$ -ig sorszámozva. A versenybíró minden fordulóban egy  $S$  és  $E$  pozíciót ad meg ( $0 \leq S < E \leq N - 1$ ). A fordulóban az  $S$ . és  $E$ . pozíció közötti lovagok vesznek részt. A győztes visszaáll az eredeti pozíciójára és folytatja a versenyt, míg a többiek kiesnek és elhagyják a helyüket. Ezután a megmaradt lovagok egymás mellé sorakoznak a  $0 \dots N - (E - S) - 1$  pozíción, megtartva eddigi sorrendjüket. A versenybíró a fordulókat addig folytatja, amíg egyetlen lovag marad.

Leonardo tudja, hogy a lovagok különböző erősségűek, a leggyengébb 0 értékű, a legerősebb pedig  $N-1$ , minden fordulóban a legerősebb résztvevő a győztes. Azt is tudja, hogy a versenybíró  $C$  fordulót rendez mely pozíción álló lovagokkal.

### Kései lovag

$N-1$  lovag felsorakozott, de a legnépszerűbb hiányzik. A hiányzó erőssége  $R$ . Leonardo olyan pozíciót akar választani neki, hogy a lehető legtöbb fordulóban győzzön. Nem érdekes, hogy ki a győztes azokban a fordulóban, amelyekben nem vesz részt a késő lovag.

### Példa

$N=5$ , a többi  $N-1$  lovag erőssége:  $[1, 0, 2, 4]$ , azaz a késő lovag erőssége 3.  $C=3$  forduló lesz, a versenybíró a 3 fordulóban az  $(1, 3)$ ,  $(0, 1)$ ,  $(0, 1)$  pozíciókban állókat szólítja versenybe.

Ha Leonardo az első pozícióra helyezi a késő lovagot, akkor az erősségek:  $[3, 1, 0, 2, 4]$ . Az első fordulóban az 1,2,3 pozíción levő 1,0,2 erősségű lovagok vesznek részt, a 2 erősségű győz és visszaáll a helyére. z új erőssorrend:  $[3, 2, 4]$ . A következő fordulóban az  $R=3$  győz a 2 erősségű ellen, az új erőssorrend:  $[3, 4]$ . Az utolsó fordulóban a 4 győz. Így a késő lovag csak egyszer győzött, a második fordulóban.

Ha azonban Leonardo az 1 és 0 erősségű közé állítja a késő lovagot, akkor az erőssorrend:  $[1, 3, 0, 2, 4]$ . Az első fordulóban a 3,0,2 erősségűek vesznek részt, az  $R=3$  győz. Az új erősségsorrend:  $[1, 3, 4]$ . A második fordulóban az  $R=3$  legyőzi az 1 erősségűt. A végső erősségsorrend:  $[3, 4]$ , győz a 4. Így a késő lovag kétszer győzött, ilyen feltételek mellett ez a lehető legjobb.

## Feladat

Írj programot, amely meghatározza a késő lovag legjobb pozícióját, ahol a legtöbb győzelmet arathatja. A feladat pontosan a `GetBestPosition(N, C, R, K, S, E)` eljárás megírása, ahol:

- $N$  a lovagok száma;
- $C$  a fordulók száma ( $1 \leq C \leq N - 1$ );
- $R$  a késő lovag erőssége,  $0$  és  $N-1$  közötti érték, különbözik a sorban álló lovagok erősségétől (a  $K$  tömbből is kikövetkeztethető);
- $K$  egy  $N-1$  egész számot tartalmazó tömb, a sorban állók erősségei;
- $S$  és  $E$  egy-egy  $C$  elemű tömb,  $0$ -tól  $C-1$ -ig indexelve, az egyes fordulók kezdő és végpozícióit adja meg, ahol  $S[i] < E[i]$ .

A paraméterek biztosan helyesek,  $E[i]$  mindig kisebb, mint az  $i+1$ . fordulóban részt vevő lovagok száma, továbbá a  $C$ . forduló után csak  $1$  lovag marad.

A `GetBestPosition(N, C, R, K, S, E)` adja vissza a késő lovag legjobb  $P$  pozícióját ( $0 \leq P \leq N - 1$ )! Több megoldás esetén a legkisebbet kell kiírni.  $P$  az a szám, ahány lovag a késő lovag előtt áll az első forduló előtt. Azaz pl.  $P=0$  azt jelenti, hogy a késő lovag az első helyre kerül,  $P=N-1$  pedig azt, hogy az utolsóra.

### 1. részfeladat [17 pont]

$N \leq 500$ .

### 2. részfeladat [32 pont] =

$N \leq 5\,000$ .

### 3. részfeladat [51 pont] =

$N \leq 100\,000$ .

# Megvalósítás

Egyetlen file-t kell beküldened: `tournament.c`, `tournament.cpp` vagy `tournament.pas`. Ebben kell megvalósítanod a `GetBestPosition` eljárást.

## C/C++ program

```
int GetBestPosition(int N, int C, int R, int *K, int *S, int *E);
```

## Pascal program

```
function GetBestPosition(N, C, R : LongInt; var K, S, E : array of LongInt) : LongInt;
```

Írhatsz más eljárásokat is, nem használhatod a standard inputot, outputot és file-okat sem.

## Minta értékelő

A minta értékelő a következő formában várja az adatokat:

- 1. sor: N, C, R;
- 2, ..., N. sorok: K[i];
- N + 1, ..., N + C + 1. sorok: S[i], E[i].

## Idő és memória limitek

- Időlimit: 1 másodperc.
- Memórialimit: 256 MiB.