

Turnirul

În anul 1491 ducele de Milano Lodovico I-a rugat pe Leonardo să-i orchestreze nunta sa cu Beatrice d'Este, care includea și un mare turneu, ce trebuia să dureze trei zile. Dar cel mai vestit cavaler întârzia ...

Turnirul

Într-un turneu, N cavaleri sunt mai întâi aranjați într-o linie, iar pozițiile lor de-a lungul liniei sunt numerotate de la 0 la $N - 1$. Arbitrul turneului stabilește o rundă cerând să iasă din linie cavalerii din pozițiile dintre S și E inclusiv (unde $0 \leq S < E \leq N - 1$). Toți cavalerii de pe pozițiile între S și E (inclusiv) se luptă între ei. Câștigătorul acestei lupte se întoarce la locul său din linie și rămâne în turneu, în timp ce perdanții iese de pe câmp și sunt eliminați din turneu. În continuare, cavalerii rămași se aranjează de-a lungul liniei, de la începutul ei, păstrând ordinea lor relativă din linia precedentă, ocupând astfel pozițiile de la 0 la $N - (E - S) - 1$. În continuare, arbitrul stabilește runda următoare, repetând acest proces până când rămâne doar un singur cavaler.

Leonardo cunoaște faptul că toți cavalerii au puteri diferite, reprezentate prin ranguri distincte de la 0 (cel mai slab) la $N - 1$ (cel mai puternic). De asemenea, el cunoaște exact ce echipe vor fi desemnate de arbitru pentru cele C runde. La urma urmei el e Leonardo, ... și el este sigur că în fiecare din aceste runde va câștiga cavalerul cu rangul cel mai înalt.

Cavalerul ce a întârziat

Cei $N - 1$ din totalul de N cavaleri sunt deja aranjați în linie. Lipsește doar cavalerul cel mai vestit. Acest cavaler are rangul R și vine la turnir un pic mai târziu. Pentru a crea o atmosferă de divertisment, Leonardo vrea să exploateze popularitatea cavalerului întârziat și alege pentru el o astfel de poziție în linie, care va maximiza numărul de runde pe care cavalerul întârziat le va câștiga. Rețineți că nu suntem interesați în rundele în care nu participă cavalerul întârziat, ci doar în rundele la care el ia parte și le câștigă.

Exemplu

Pentru $N = 5$, cei $N - 1$ cavaleri aranjați în linie au rangurile $[1, 0, 2, 4]$, respectiv. Evident, rangul cavalerului întârziat este $R = 3$. Pentru fiecare din cele $C = 3$ runde, arbitrul intenționează să scoată din linie cavalerii de pe pozițiile (S, E) , în următoarea ordine: $(1, 3)$, $(0, 1)$, $(0, 1)$.

Dacă Leonardo plasează cavalerul întârziat în prima poziție, rangurile cavalerilor de pe linie vor fi $[3, 1, 0, 2, 4]$. În prima rundă se vor lupta cavalerii de pe pozițiile 1, 2, 3, cu rangurile 1, 0, 2, iar învingător va fi cavalerul cu rangul 2. Cavalerii din linia nouă vor avea rangurile $[3, 2, 4]$. În următoarea rundă se vor lupta cavalerii cu rangurile 3 și 2 (din pozițiile 0, 1), câștigător va fi

cavalerul cu rangul $R = 3$. După această rundă rangurile cavalerilor din linie vor fi $[3, 4]$. Runda finală (cavalerii de pe pozițiile 0, 1) este câștigată de cavalerul cu rangul 4. Prin urmare, cavalerul întârziat câștigă doar o rundă (a doua).

În cazul în care Leonardo ar plasa cavalerul întârziat între cavalerii cu rangurile 1 și 0, linia ar arăta în felul următor: $[1, 3, 0, 2, 4]$. De data aceasta, în prima rundă participă cavalerii cu rangurile 3, 0, 2, câștigător fiind cavalerul cu rangul $R = 3$. Linia următoare este formată din cavalerii cu rangurile $[1, 3, 4]$. În runda următoare participă cavalerii cu rangurile (1 și 3), din nou câștigător fiind cel cu rangul $R = 3$. Linia finală este formată din cavalerii cu rangurile $[3, 4]$, câștigător fiind cel cu rangul 4. Astfel, cavalerul întârziat câștigă două runde; de fapt, acesta este cel mai bun plasament posibil, deoarece nu există nici o altă modalitate de plasare în care cavalerul întârziat ar câștiga mai mult decât de două ori.

Sarcină

Trebuie să scrii un program care alege cea mai bună poziție pentru cavalerul întârziat, astfel încât numărul de runde castigate de el să fie maxim, așa cum și-o dorește Leonardo. Mai exact, trebuie să elaborezi o rutină denumită `GetBestPosition` (N, C, R, K, S, E), unde:

- N este numărul de cavaleri;
- C este numărul de runde organizate de arbitru ($1 \leq C \leq N - 1$);
- R este rangul cavalerului întârziat; rangurile tuturor cavalerilor (atât a celor care sunt deja în linie, cât și al celui întârziat) sunt numere distincte din intervalul $0, \dots, N - 1$; cu toate că rangul R al cavalerului întârziat poate fi dedus, el este indicat explicit;
- K este un tablou din $N - 1$ numere întregi, reprezentând rangurile celor $N - 1$ cavaleri aranjați deja în linia inițială;
- S și E sunt două tablouri de dimensiunea C . Pentru orice i de la 0 până la $C - 1$, inclusiv, runda $(i + 1)$ formată de arbitru va include toți cavalerii de pe pozițiile $S[i]$ până la $E[i]$, inclusiv. Pentru fiecare i , $S[i] < E[i]$.

Toate apelurile acestei rutine vor fi corecte: $E[i]$ va fi mai mic decât numărul curent de cavaleri rămași pentru runda $(i + 1)$, iar după C runde va rămâne exact un singur cavaler.

`GetBestPosition`(N, C, R, K, S, E) trebuie să returneze cea mai bună poziție P în care Leonardo trebuie să-l plaseze pe cavalerul întârziat ($0 \leq P \leq N - 1$). Dacă există mai multe poziții echivalente, retunați poziția cea mai mică. (Poziția P este o poziția 0-bazată a cavalerului întârziat în linia rezultat. Prin alte cuvinte, P este numărul celorlați cavaleri care în soluția optimală stau înaintea cavalerului întârziat. În particular, $P = 0$ semnifică faptul că cavalerul întârziat este la începutul liniei, iar $P = N - 1$ semnifică faptul că el este la sfârșitul liniei).

Subtask 1 [17 puncte]

Se consideră că $N \leq 500$.

Subtask 2 [32 de puncte]

Se consideră că $N \leq 5\,000$.

Subtask 3 [51 de puncte]

Se consideră că $N \leq 100\,000$.

Detalii de implementare

Trebuie să transmiți exact un singur fișier, denumit `tournament.c`, `tournament.cpp` sau `tournament.pas`. Acest fișier trebuie să implementeze subprogramul descris mai sus, având următoarele semnături:

Programele C/C++

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

Programele Pascal

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

Aceste subprograme trebuie să se comporte așa cum este descris mai sus. Desigur, pentru uzul intern ești liber să pui în aplicare și alte subprograme. Submit-urile tale nu trebuie să interacționeze în nici un fel cu intrarea/ieșirea standard și cu alte fișiere.

Model de evaluator

Modelul de evaluator furnizat de mediul de evaluare acceptă următorul format al datelor de intrare:

- linia 1: N, C, R ;
- liniile 2, ..., N : $K[i]$;
- liniile $N + 1$, ..., $N + C$: $S[i], E[i]$.

Restricții pentru timpul de execuție și memoria folosită

- Limita de timp: 1 secundă.
- Limita de memorie: 256 MiB.