

Riddertoernooi

De hertog van Milaan, Ludovico Sforza, trouwde in 1491 met Beatrice d'Este. Leonardo werd gevraagd om de feestelijkheden te organiseren, waaronder een groot riddertoernooi dat 3 dagen duurde. Maar naar goede Italiaanse gewoonte is de meest populaire ridder te laat...

Toernooi

In het riddertoernooi worden N ridders op een rij gezet. Hun posities zijn genummerd van 0 tot en met $N - 1$. De toernooimeester start een *ronde* door twee posities S en E af te roepen (waarbij $0 \leq S < E \leq N - 1$). Alle ridders met posities van S tot en met E treden met elkaar in het strijdperk. De winnaar gaat naar de volgende ronde en gaat terug naar zijn plek in de rij. De verliezers zijn uitgeschakeld en verlaten het veld. Daarna schuiven de ridders op naar het begin van de rij, waarbij ze hun volgorde behouden, zodat hun posities nu genummerd zijn van 0 tot en met $N - (E - S) - 1$. De toernooimeester start dan een volgende ronde, en herhaalt dit proces totdat slechts één ridder overblijft.

Leonardo weet dat geen twee ridders even sterk zijn, en hun sterkte wordt aangegeven met een graad van 0 (de zwakste) tot en met $N-1$ (de sterkste). Hij weet op voorhand de precieze opdrachten die de toernooimeester zal geven voor de C rondes. Hij weet ook zeker dat in iedere ronde de ridder met de hoogste graad (de sterkste) wint.

De late ridder

$N-1$ van de N ridders staan al in de rij. Alleen de meest populaire ridder ontbreekt nog. Deze ridder heeft sterkte R en komt wat later. Om het publiek te amuseren wil Leonardo de populariteit van deze ridder gebruiken, en hem zo in de rij plaatsen dat hij zo veel mogelijk rondes wint. Let op: we zijn niet geïnteresseerd in de rondes waarin deze ridder niet meedoet, enkel in de rondes waarin hij wel meedoet en wint.

Voorbeeld

Voor $N = 5$ ridders, hebben de $N - 1$ ridders die al in de rij staan de sterktes $[1, 0, 2, 4]$. Hieruit volgt al dat de late ridder sterkte $R = 3$ heeft. De toernooimeester wil $C = 3$ rondes organiseren en wil voor elke ronde posities (S, E) afroepen, in deze volgorde: $(1, 3)$, $(0, 1)$, $(0, 1)$.

Als Leonardo de late ridder vooraan zet, dan zijn de sterktes van de de ridders in de rij achtereenvolgens $[3, 1, 0, 2, 4]$. De eerste ronde bevat dan ridders (op posities 1, 2 en 3) met sterktes 1, 0 en 2: de ridder met sterkte 2 is de winnaar. De nieuwe rij is $[3, 2, 4]$. De volgende ronde gaat tussen ridders met sterkte 3 en 2 (posities 0 en 1) en die met sterkte $R = 3$ wint. De rij is nu $[3, 4]$. De laatste ronde (tussen posities 0 en 1) heeft 4 als winnaar. De late ridder wint zo slechts

één ronde (de tweede).

Als Leonardo daarentegen de late ridder tussen die met sterktes 1 en 0 zet, ziet de rij er als volgt uit: [1, 3, 0, 2, 4]. De eerste ronde gaat nu tussen sterktes 3, 0 en 2 en de ridder met sterkte $R = 3$ wint. Het resultaat is de rij [1, 3, 4] en in de volgende ronde (1 tegen 3) wint de ridder met $R = 3$ opnieuw. De laatste rij is [3, 4] en 4 wint. De late ridder wint zo twee rondes: dit is zelfs de best mogelijke plek om hem in de rij te zetten omdat de late ridder op geen enkele manier meer dan twee rondes kan winnen.

Opdracht

Schrijf een programma dat de beste positie voor de late ridder bepaalt zodat het aantal rondes dat hij wint maximaal is. Specifiek: je moet een routine `GetBestPosition(N, C, R, K, S, E)` implementeren, waarbij:

- N is het aantal ridders;
- C is het aantal rondes dat de toernooimeester zal afroepen ($1 \leq C \leq N - 1$);
- R is de sterkte van de late ridder. De sterktes van alle ridders (die in de rij plus de late ridder) zijn onderling verschillend en gekozen uit 0 tot en met $N - 1$. De sterkte R van de late ridder wordt expliciet gegeven, ook al kan ze eigenlijk afgeleid worden;
- K is een array van $N - 1$ integers, die de sterktes voorstellen van de $N - 1$ ridders die al in de rij staan;
- S and E zijn twee arrays van grootte C : voor elke i tussen 0 en $C - 1$ (inclusief), zal de $(i+1)$ ste ronde die door de toernooimeester wordt afgeroepen, alle ridders bevatten van positie $S[i]$ tot en met positie $E[i]$. Je mag aannemen dat voor elke i , $S[i] < E[i]$.

Elke aanroep van deze routine is geldig: $E[i]$ is altijd lager dan het aantal ridders dat overblijft in de $(i+1)$ ste ronde, en na de C functie-aanroepen zal er precies één ridder overblijven.

`GetBestPosition(N, C, R, K, S, E)` moet de beste positie P teruggeven waarop Leonardo de late ridder moet zetten ($0 \leq P \leq N - 1$). Als er meerdere gelijkwaardige posities zijn, *geef dan de kleinste terug*. (De positie P is de positie van de late ridder in de resulterende rij, die begint bij positie 0. Met andere woorden: P is het aantal ridders dat in rij staat vóór de late ridder in de optimale oplossing. Dus, $P = 0$ betekent dat de ridder helemaal vooraan in de rij komt en $P = N - 1$ betekent dat hij helemaal achteraan komt.)

Subtaak 1 [17 punten]

Je mag ervan uitgaan dat $N \leq 500$.

Subtaak 2 [32 punten]

Je mag ervan uitgaan dat $N \leq 5\,000$.

Subtaak 3 [51 punten]

Je mag ervan uitgaan dat $N \leq 100\,000$.

Implementatiedetails

Je moet exact één bestand indienen, genaamd `tournament.c`, `tournament.cpp` of `tournament.pas`. Dit programma moet de hierboven beschreven functie implementeren volgens de volgende declaratie:

C/C++ programma's

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

Pascal programma's

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

Deze functie moet werken zoals hierboven werd beschreven. Natuurlijk staat het je vrij bijkomende functies te implementeren voor intern gebruik. Je code mag op geen enkele manier interageren met standaard input/output, of met andere bestanden.

Voorbeeld-graders

De voorbeeld-grader die je krijgt in de testomgeving verwacht invoer in het volgende formaat:

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

Tijds- en geheugenlimieten

- Tijdslimiet: 1 seconde.
- Geheugenlimiet: 256 MiB.