

Joutes (Jousting tournament)

Pour son mariage avec Béatrice d'Este en 1491, le duc de Milan, Lodovico Sforza, a demandé à Léonard d'animer sa cérémonie de mariage comportant une prestigieuse joute de trois jours. Cependant, le chevalier le plus célèbre est en retard...

Tournoi

On commence une joute en alignant les N chevaliers sur une ligne et en numérotant leurs positions de 0 à $N - 1$. Le maître de joute lance un *tour* en donnant deux positions S et E (où $0 \leq S < E \leq N - 1$). Tous les chevaliers dont les positions sont entre S et E (inclus) combattent : le vainqueur continue le tournoi et retourne à sa place sur la ligne tandis que les perdants sont éliminés et quittent le tournoi. Ensuite, les chevaliers restants sont regroupés en début de ligne tout en gardant leur ordre relatif, de telle façon que leurs nouvelles positions aillent de 0 à $N - (E - S) - 1$ (inclus). Le maître de joute lance ensuite un autre tour, et répète ce processus jusqu'à ce qu'il n'y ait plus qu'un seul chevalier.

Léonard sait que tous les chevaliers ont des niveaux différents, représentés sous la forme de rangs distincts allant de 0 (le plus faible) à $N - 1$ (le plus fort). Il connaît également exactement les commandes que le maître de joute lors des C tours, c'est Léonard après tout... et il est certain que lors de chacun de ces tours, le chevalier avec le plus haut rang va gagner.

Le chevalier retardataire

Alors que $N - 1$ des N chevaliers sont déjà alignés, le plus célèbre chevalier manque à l'appel. Ce chevalier a un rang R et arrive un peu en retard. Afin de rendre le spectacle le plus intéressant possible, Léonard désire profiter de sa popularité et lui choisir une position dans la ligne qui optimisera le nombre de tours que le chevalier célèbre gagnera. Notez que nous ne sommes pas intéressés par les tours qui n'impliquent pas le chevalier retardataire, mais uniquement par ceux auxquels il prend part et gagne.

Exemple

Pour $N = 5$ chevaliers, les $N - 1$ chevaliers qui sont déjà alignés ont respectivement les rangs $[1, 0, 2, 4]$. Le chevalier retardataire a, lui, le rang $R = 3$. Pour les $C = 3$ tours, le maître de joute compte appeler les positions (S, E) à chaque tour dans cet ordre: $(1, 3)$, $(0, 1)$, $(0, 1)$.

Si Léonard insère le chevalier retardataire à la première position, les rangs des chevaliers sur la ligne seront $[3, 1, 0, 2, 4]$. Le premier tour implique les chevaliers (aux positions $1, 2, 3$) avec les rangs $1, 0, 2$, permettant au chevalier au rang 2 de gagner. La nouvelle ligne est donc $[3, 2, 4]$. Le tour suivant est 3 contre 2 (aux positions 0 et 1), et le chevalier avec le rang $R = 3$ gagne, ce qui

permet d'obtenir la ligne [3, 4]. Le dernier tour (aux positions 0 et 1) voit la victoire du 4. Le chevalier retardataire ne gagne donc qu'un tour (le second).

Si, par contre, Léonard insère le chevalier retardataire entre ceux de rangs 1 et 0, la ligne sera [1, 3, 0, 2, 4]. Dans ce cas, le premier tournoi implique 3, 0 et 2, et le chevalier de rang $R = 3$ gagne. Ensuite, la ligne est [1, 3, 4] et dans le tour suivant (1 contre 3), le chevalier de rang $R = 3$ gagne de nouveau. La ligne finale est [3, 4] et 4 gagne. Le chevalier retardataire gagne ainsi deux tours, ce qui est le meilleur placement possible puisqu'il n'a pas la possibilité de gagner plus de deux fois.

Problème

Votre tâche est d'écrire un programme qui choisit la meilleure position pour le chevalier retardataire afin qu'il gagne le plus de tours possible, comme le souhaite Léonard. En pratique, vous devez implémenter une fonction appelée `GetBestPosition(N, C, R, K, S, E)`, dans laquelle :

- N est le nombre de chevaliers ;
- C est le nombre de tours appelés par le maître de joute ($1 \leq C \leq N - 1$) ;
- R est le rang du chevalier retardataire. Le rang de chacun des chevaliers (ceux déjà alignés ou celui retardataire) est différent des autres et compris entre 0 et $N - 1$ (inclus). Le rang R du chevalier retardataire est donné explicitement bien qu'il pourrait être déduit ;
- K est un tableau de $N - 1$ entiers, représentant le rangs des $N - 1$ chevaliers alignés au départ ;
- S et E sont deux tableaux de taille C : pour tout i compris entre 0 et $C - 1$ (inclus), le $(i + 1)$ ème tour appelé par le maître de joute impliquera tous les chevaliers entre les positions $S[i]$ et $E[i]$, inclus. Vous pouvez considérer que pour tout i , $S[i] < E[i]$.

Les appels à cette fonction seront toujours valides : $E[i]$ sera plus petit que le nombre courant de chevaliers restants pour le $(i + 1)$ ème tour et après les C commandes, il ne restera qu'un seul chevalier.

`GetBestPosition(N, C, R, K, S, E)` doit retourner la meilleure position P à laquelle Léonard devrait placer le chevalier retardataire ($0 \leq P \leq N - 1$). S'il y a plusieurs positions équivalentes, *retournez la position dont la valeur est la plus petite*. (La position P est, si on commence à l'index 0, la position du chevalier retardataire de la ligne résultante. En d'autre mots, P est le nombre d'autres chevaliers qui se trouvent avant le chevalier en retard dans la solution optimale. Plus précisément, $P = 0$ signifie que le chevalier retardataire est placé en début de ligne et $P = N - 1$ signifie qu'il est placé en dernière position.)

Sous-tâche 1 [17 points]

Vous pouvez considérer que $N \leq 500$.

Sous-tâche 2 [32 points]

Vous pouvez considérer que $N \leq 5\,000$.

Sous-tâche 3 [51 points]

Vous pouvez considérer que $N \leq 100\,000$.

Détails d'implémentation

Vous devez soumettre un seul fichier, appelé `tournoiement.c`, `tournoiement.cpp` ou `tournoiement.pas`. Ce fichier doit implémenter la fonction décrite ci-dessus avec les signatures suivantes.

Programmes C/C++

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

Programmes Pascal

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

Ces fonctions doivent se comporter comme décrit ci-dessus. Bien entendu, vous pouvez implémenter d'autres fonctions pour un usage interne. Vos soumissions ne peuvent pas interagir de quelque façon que ce soit avec l'entrée/sortie standard ni avec quelconque autre fichier.

Évaluateurs d'exemple

L'évaluateur d'exemple fourni avec la tâche attend une entrée dans le format suivant:

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

Limites de temps et de mémoire

- Limite de temps : 1 seconde
- Limite de mémoire : 256 Mio