



馬上試合大会 (Jousting tournament)

1491年のベアトリーチェ・デステ (Beatrice d'Este) との結婚式のために、ミラノ公ルドヴィーコ・スフォルツァ (the Duke of Milan Lodovico Sforza) は、レオナルド (Leonardo) に結婚祝賀会の行事の企画を依頼した。そこでは馬上試合大会が行われ、丸三日間続いたという。しかし、最も人気のある騎士は遅刻してしまい…

大会 (Tournament)

馬上試合大会では、初め N 人の騎士が直線上に並び、その場所には順に 0 から $N - 1$ までの番号がつけられている。馬上試合の主催者は 2 つの場所 S と E ($0 \leq S < E \leq N - 1$) を宣言し、ラウンド (round) を行う。場所 S から E まで (S と E を含む) にいるすべての騎士が戦う：勝った 1 人の騎士は元の場所に戻って大会を続け、負けた騎士たちは敗退し競技場を離れる。その後、残った騎士は相対的な順番を保ったまま場所 0 から $N - (E - S) - 1$ までに並び直す。主催者は再びラウンドを行い、これをちょうど 1 人の騎士が残るまで繰り返す。

レオナルドはすべての騎士が異なる強さを持っていることを知っている。強さは 0 (もっとも弱い) から $N - 1$ (最も強い) までの異なる値で表される。彼は C 回のラウンドで主催者が宣言する S と E を知っている。…何せ主催者はレオナルドだから。また、彼は各ラウンドでは、最も強い騎士が勝つと確信している。

遅刻した騎士 (Late knight)

最も人気のある騎士を除いた $N - 1$ 人の騎士は既に直線上に並んでいる。最も人気のある騎士は強さ R を持ち、少し遅刻して到着する。遅刻した騎士の人気を利用して大会を盛り上げるため、レオナルドはその騎士が勝つラウンドの回数を最大にできる場所を知りたい。遅刻した騎士が戦わないラウンドは数えず、戦って勝ったラウンドのみを数えることに注意せよ。

例 (Example)

騎士の人数は $N = 5$ であり、既に並んでいる $N - 1$ 人の騎士の強さはそれぞれ $[1, 0, 2, 4]$ であるとする。遅刻した騎士の強さは $R = 3$ となる。ラウンドの回数は $C = 3$ であり、主催者の宣言する (S, E) の値は順に $(1, 3)$, $(0, 1)$, $(0, 1)$ であるとする。

レオナルドが遅刻した騎士を最初の場所に入れると、騎士の強さは $[3, 1, 0, 2, 4]$ となる。最初のラウンドには場所 $1, 2, 3$ にいる強さ $1, 0, 2$ の騎士が参加し、強さ 2 の騎士が勝ち残る。新しい騎士の列は $[3, 2, 4]$ である。次のラウンドには場所 $0, 1$ にいる強さ $3, 2$ の騎士が参加し、強さ $R = 3$ の騎士が勝ち、騎士の列は $[3, 4]$ となる。最後のラウンドには場所 $0, 1$ にいる強さ $3, 4$ の騎士が参加し、強さ 4 の騎士が勝つ。よって、遅刻した騎士は 1 回のラウンド (2 回目のラウンド) だけで勝つ。

代わりに、レオナルドが遅刻した騎士を強さ 1 と 0 の騎士の間に入れると、騎士の強さは [1, 3, 0, 2, 4] となる。最初のラウンドには強さ 3, 0, 2 の騎士が参加し、強さ $R = 3$ の騎士が勝つ。新しい騎士の列は [1, 3, 4] である。次のラウンドには強さ 1, 3 の騎士が参加し、強さ $R = 3$ の騎士が再び勝つ。最後の騎士の列は [3, 4] であり、強さ 4 の騎士が勝つ。よって、遅刻した騎士は 2 回のラウンドで勝つ。遅刻した騎士が 2 回より多く勝つ方法は存在しないので、これは最適な配置である。

問題文 (Statement)

あなたの課題は、レオナルドが望むように、遅刻した騎士が勝つラウンドの回数が最大になるような場所を選ぶプログラムを書くことである。具体的には、あなたはルーチン `GetBestPosition(N, C, R, K, S, E)` を実装しなければならない。引数は以下の通りである：

- N は騎士の人数である。
- C は主催者が行うラウンドの回数である ($1 \leq C \leq N - 1$)。
- R は遅刻した騎士の強さである。騎士 (既に並んでいる者も遅刻した者も含む) の強さは 0 以上 $N - 1$ 以下の相異なる値で表される。 R は並んでいる騎士の強さから求めることができるが、直接与えられる。
- K は $N - 1$ 個の整数の配列であり、既に並んでいる $N - 1$ 人の騎士の強さを表す。
- S と E はサイズ C の配列である。0 以上 $C - 1$ 以下の各 i に対し、 $(i + 1)$ 回目のラウンドには場所 $S[i]$ から $E[i]$ まで ($S[i]$ と $E[i]$ を含む) にいるすべての騎士が参加する。各 i に対し、 $S[i] < E[i]$ であると仮定してよい。

このルーチンに渡される S と E は正当な値をもつ： $E[i]$ は $(i + 1)$ 回目のラウンドが行われる前段階で残っている騎士の人数より小さく、 C 回のラウンドの後にはちょうど 1 人の騎士が残る。

`GetBestPosition(N, C, R, K, S, E)` は、レオナルドが遅刻した騎士を配置する最もよい場所 P ($0 \leq P \leq N - 1$) を返さなければならない。最もよい場所が複数あるときは、そのうち最も小さい場所を返さなければならない。 P は遅刻した騎士の、追加された後の列での場所を表し、0 から数える。すなわち、 P は最適解において遅刻した騎士より前にいる騎士の人数である。例えば、 $P = 0$ は遅刻した騎士が列の最初に入ること、 $P = N - 1$ は列の最後に入ることを示す。

小課題 1 [17 点]

$N \leq 500$ と仮定してよい。

小課題 2 [32 点]

$N \leq 5\,000$ と仮定してよい。

小課題 3 [51 点]

$N \leq 100\,000$ と仮定してよい.

実装の詳細 (Implementation details)

あなたは `tournament.c`, `tournament.cpp` または `tournament.pas` という名前のファイルをちょうど 1 つ提出しなければならない. そのファイルは上で説明されたサブプログラムを以下のシグネチャを用いて実装しなければならない.

C/C++ プログラム

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

Pascal プログラム

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

このサブプログラムは上記で説明された通りに動作しなければならない. もちろん, 内部での使用のために他のサブプログラムを実装することは自由である. あなたの提出は標準入力・標準出力, あるいは他のファイルといかなる方法でもやりとりしてはならない.

採点プログラムのサンプル (Sample graders)

課題環境に与えられる採点プログラムのサンプルは以下の書式の入力を読み込む:

- 1 行目: N, C, R .
- 2 行目から N 行目まで: $K[i]$.
- $N + 1$ 行目から $N + C$ 行目まで: $S[i], E[i]$.

時間とメモリの制限 (Time and Memory limits)

- 時間制限: 1 秒.
- メモリ制限: 256 MiB.