

## Torneo de Justas

Para su boda con Beatrice d'Este en 1491, el Duque de Milán Lodovico Sforza pidió a Leonardo orquestar la celebración de la boda, incluyendo un buen torneo de justas que durara por tres días enteros. Pero el caballero más popular se ha demorado...

### Torneo

En un torneo de justas, los  $N$  caballeros son primeramente acomodados en una línea y sus posiciones son numeradas del 0 al  $N - 1$  siguiendo el orden de la línea. El maestro de la justa configura un round seleccionando dos posiciones  $S$  y  $E$  (donde  $0 \leq S < E \leq N - 1$ ). Todos los caballeros cuyas posiciones estén entre  $S$  y  $E$  (inclusive) compiten: El ganador continúa en el torneo y vuelve a su lugar en la línea, mientras que los perdedores salen del juego y abandonan el campo. Después de eso, los caballeros que quedan se juntan en una nueva fila manteniendo su orden relativo en la línea, entonces sus posiciones resultantes son desde 0 hasta  $N - (E - S) - 1$ . El maestro de la justa llama a otro round, repitiendo este proceso hasta que queda un solo caballero.

Leonardo sabe que todos los caballeros tienen diferentes fortalezas, representadas por distintas números de 0 (el peor) hasta  $N - 1$  (el mejor). Además él sabe los llamados exactos que el maestro de la justa va a hacer para los  $C$  rounds: él es Leonardo, después de todo... y él esta seguro que en cada uno de esos rounds el caballero con la fortaleza más alta va a ganar.

### El Caballero Demorado

$N - 1$  de los  $N$  caballeros han ya llegado y están acomodados en la línea, solo falta el caballero más popular. Este caballero tiene una fortaleza  $R$  y va a llegar un poco más tarde. Para beneficio del entretenimiento, Leonardo busca explotar su popularidad y elegir para él una posición en la línea de modo que maximice el número de encuentros en los que el caballero demorado gana. Nota que no estamos interesados en los rounds que no incluyen al caballero demorado, solo en los que participa y gana.

### Ejemplo

Para  $N = 5$  caballeros, los  $N - 1$  caballeros que están ya acomodados en la línea tienen fortalezas de  $[1, 0, 2, 4]$ , respectivamente. Consecuentemente, el caballero demorado tiene una fortaleza  $R = 3$ . Para los  $C = 3$  rounds, el maestro de la justa pretende llamar las  $(S, E)$  posiciones por round, en el siguiente orden:  $(1,3)$ ,  $(0,1)$ ,  $(0,1)$ .

Si Leonardo inserta al caballero demorado en la primera posición, las fortalezas de los caballeros en la línea serían [3, 1, 0, 2, 4]. El primer round involucra a los caballeros (en las posiciones 1, 2, 3) con fortalezas 1, 0, 2 respectivamente, dejando al caballero con fortaleza 2 como ganador. La nueva línea es [3, 2, 4].

En vez, si Leonardo inserta al caballero demorado en entre los caballeros con fortalezas 1 y 0, la línea quedaría como sigue: [1, 3, 0, 2, 4]. Esta vez, el primer round involucra 3, 0, 2 y el caballero con fortaleza  $R = 3$  gana. La siguiente línea es [1, 3, 4], y en el siguiente round (1 contra 3) el caballero con fortaleza  $R = 3$  gana otra vez. La línea final es [3,4], donde 4 gana. Esto es, el caballero demorado gana dos rounds: este es el mejor lugar posible dado que no hay forma de que el caballero demorado gané más de dos veces.

## Problema

Tu tarea consiste en escribir un programa que elija el mejor lugar para el caballero demorado de modo que el numero de rounds ganados por él sea maximizado. Específicamente, tienes que implementar una función llamada `GetBestPosition(N, C, R, K, S, E)`, donde:

- $N$  es el numero de caballeros;
- $C$  es el numero de rounds llamados por el maestro de la justa ( $1 \leq C \leq N - 1$ );
- $R$  es la fortaleza del caballero demorado; las fortalezas de todos los caballeros (incluidos aquellos de los que ya están formados y el demorado) son distintas y son elegidas de entre 0, ...,  $N - 1$ , la fortaleza  $R$  del caballero demorado está dado explícitamente aún si puede ser deducido.
- $K$  es un arreglo de  $N - 1$  enteros, representando las fortalezas de los  $N - 1$  caballeros que ya están formados en la línea.
- $S$  y  $E$  son dos arreglos de tamaño  $C$ , por cada  $i$  entre 0 y  $C - 1$ , inclusive, el  $(i + 1)$  ésimo round llamado por el maestro de la justa involucrará todos los caballeros desde la posición  $S[i]$  hasta la posición  $E[i]$ , inclusive. Debes asumir que por cada  $i$ ,  $S[i] < E[i]$ .

Las llamadas pasadas a esta rutina son validas, tenemos que  $E[i]$  es más pequeño que el numero actual de caballeros existentes para el  $(i + 1)$  ésimo round, y después de los  $C$  llamados solo quedará un caballero.

`GetBestPosition(N, C, R, K, S, E)` debe regresar la mejor posición  $P$  donde Leonardo debe poner al caballero demorado ( $0 \leq P \leq N - 1$ ). Si hay múltiples posiciones equivalentes, *regresa la más pequeña*. (La posición  $P$  es la posición en base 0 para el caballero demorado en la línea resultante. En otras palabras,  $P$  es el número de caballeros formados antes del caballero demorado en la mejor solución. Específicamente,  $P = 0$  significa que el caballero demorado está al inicio de la línea, y  $P = N - 1$  significa que el caballero demorado se encuentra al final.)

## Sub-tarea 1 [17 puntos]

Puedes asumir que  $N \leq 500$ .

## Sub-tarea 2 [32 puntos]

Puedes asumir que  $N \leq 5,000$ .

## Sub-tarea 3 [51 puntos]

Puedes asumir que  $N \leq 100,000$ .

## Detalles de implementación

Debes enviar un solo archivo, llamado `tournament.c`, `tournament.cpp` o `tournament.pas`. Este archivo debe implementar la función descrita arriba usando las siguientes firmas.

### Programas en C/C++

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

### Programas en Pascal

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

Dichas funciones deberán comportarse de acuerdo a la descripción anterior. Está claro que eres libre de implementar funciones adicionales para uso interno. Tus envíos no deberán interactuar de ninguna manera con la entrada/salida estándar ni con algún archivo.

### Evaluador de ejemplo (Grader)

El evaluador de ejemplo provisto con el entorno de la tarea, esperará el siguiente formato de entrada:

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

## Límites de Tiempo y Memoria

- Límite de Tiempo: 1 segundo.
- Límite de Memoria: 256 MiB.