Torneo de justa

Para su casamiento con Beatrice d'Este en 1491, el Duke de Milán Lodovico Sforza pidió a Leonardo orquestar las celebraciones del casamiento, incluyendo un gran torneo de justa que dure tres días completos. Pero el más popular caballero está atrasado…

Torneo

En un torneo de justa, los N caballeros son primeramente dispuestos en una línea y entonces sus posiciones son numeradas desde 0 a N - 1 siguiendo el orden de la línea. El maestro de justa dispone una *ronda* designando dos posiciones S y E (donde $0 \leq S < E \leq N - 1$). Todos los caballeros cuyas posiciones están entre S y E (inclusives) compiten: el ganador continúa en el torneo y vuelve a su colocación en la línea, mientras que el perdedor queda fuera del juego y abandona el campo. Después, los restantes caballeros se desplazan hacia el comienzo de la línea, preservando su orden relativo en la línea, de modo que sus resultantes posiciones van de 0 a N - (E - S) - 1. El maestro de justa convoca para otra ronda, repitiendo este proceso hasta que quede exactamente un caballero.

Leonardo sabe que todos los caballeros tiene diferentes fuerzas, representadas por rangos distintos que van desde 0 (el más débil) hasta N - 1 (el más fuerte).También sabe los comandos exactos que el maestro de justa impondrá para las C rondas: Pues para eso es Leonardo, después de todo… y está seguro que en cada una de estos rondas el caballero con el mayor rango ganará.

Caballero atrasado

N - 1 de los N caballeros ya están dispuestos en la línea, justo el más popular caballero está ausente. Este caballero tiene rango R y está llegando un poco atrasado. Para beneficiar el entretenimiento, Leonardo desea explotar su popularidad y elegir para él una posición en la línea que maximice el número de rondas que el caballero atrasado gane. Note que no estamos interesados en las rondas que no involucran al caballero atrasado, sólo en las rondas en las cuales toma parte y gana.

Ejemplo

Para N = 5 caballeros, los N - 1 caballeros que ya están dispuestos en la línea tienen rangos [1, 0, 2, 4], respectivamente. Consecuentemente, el caballero atrasado tiene rango R = 3. Para las C = 3 rondas, el maestro de justa intenta llamar las (S, E) posiciones por ronda, en este orden: (1, 3), (0, 1), (0, 1).

Si Leonardo inserta el caballero atrasado en la primera posición, los rangos de los caballeros sobre la línea será [3, 1, 0, 2, 4]. La primera ronda involucra caballeros (en posiciones 1, 2, 3) con rangos 1, 0, 2, dejando al caballero con rango 2 como el ganador. La nueva línea es [3, 2, 4]. La siguiente ronda es 3 contra 2 (en las posiciones 0, 1), y el caballero con rango R = 3 gana, dejando la línea

[3, 4]. La ronda final (en posiciones 0, 1) tiene 4 como ganador. Entonces, el caballero atrasado solamente gana una ronda (la segunda).

En cambio, si Leonardo inserta al caballero atrasado entre los dos de rangos 1 y 0, la línea aparece como: [1, 3, 0, 2, 4]. Esta vez, la primera ronda envuelve 3, 0, 2, y el caballero con rango R = 3 gana. La siguiente línea de largada es [1, 3, 4], y en la siguiente ronda (1 contra 3) el caballero con rango R = 3 gana nuevamente. La línea final es [3, 4], donde 4 gana. Entonces, el caballero atrasado gana dos rondas: esto es actualmente la mejor colocación posible ya que no hay forma para que el caballero atrasado gane más de dos veces.

Enunciado

Tu tarea es escribir un programa que elija la mejor posición para el caballero atrasado de modo tal que el número de rondas ganadas por él es maximizado, tal como Leonardo desea. Específicamente, tienes que implementar una rutina llamada GetBestPosition(N, C, R, K, S, E), donde:

- N es el número de caballeros;
- C es el número de rondas convocadas por el maestro de justa ($1 \leq C \leq N - 1$);
- R es el rango de el caballero atrasado; los rangos de todos los caballeros (tanto de los ya alineados y del atrasado) son distintos y elegidos entre 0, …, N - 1, y el rango R del caballero atrasado es dado explicitamente no obstante que puede ser deducido;
- K es un arreglo de N - 1 enteros, representando los rangos de los N - 1 caballeros que ya están sobre the línea de arranque ;
- S y E son dos arreglos de tamaño C: para cada i entre 0 y C - 1, inclusive, la ronda i + 1 convocada por el maestro de justa involucra todos los caballeros desde la posición S[i] hasta la posición E[i], inclusives. Puedes suponer que para cada i, S[i] < E[i].

Las llamadas pasadas a esta rutina son válidas: Se asegura que E[i] es menor que el número corriente de caballeros remanentes para competir en la ronda (i+1)-ésima, y después de los C comandos restará exactamente un caballero.

`GetBestPosition`(N, C, R, K, S, E) debe devolver la mejor posición P donde Leonardo debería poner al caballero atrasado ($0 \leq P \leq N - 1$). Si hay múltiples posiciones equivalentes, *devuelva la menor*. (La posición P es la posición arrancando en 0 del caballero atrasado en la línea resultante. en otras palabras, P es el número de caballeros posicionados delante del caballero atrasado en la solución optimal. Especifically, P = 0 significa que el caballero atrasado está al comienzo de la línea, y P = N - 1 significa que está al final de ella.)

Subtarea 1 [17 puntos]

Puedes suponer que $N \leq 500$.

Subtarea 2 [32 puntos]

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

Subtarea 3 [51 puntos]

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

Detalles de implementación

Tienes que enviar exactmente un archivo, llamado `tournament.c`, `tournament.cpp` o `tournament.pas`. Este archivo debe implementar el subprograma descripto arriba usando los siguiente encabezamientos.

Programas C/C++

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

Programas Pascal

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

Estos subprogramas deben comportarse como descripto arriba. Por supuesto eres libre de implementar otros subprogramas para su uso interno. Tu envío no debe interactuar de ningún modo con la entrada/salida estándar, ni sobre ningún otro archivo.

Evaluador de muestra

El evaluador de muestra provisto en el ambiente de la tarea supondrá una entrada en el siguiente formato:

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

Límites de tiempo y memorias

- Límite de tiempo: 1 segundo.
- Límite de memoria: 256 MiB.