騎馬比武競賽

1491 年公爵 Milan Lodovico Sforza 為了他與 Beatrice d'Este 的婚禮，要求Leonardo 來負責籌備婚禮的慶典。在這個慶典中包含了一個盛大的持續三天的騎馬比武競賽，但是最受歡迎的騎士遲到了...

競賽

在一騎馬比武的競賽，N 個騎士一開始被排成一排然後按照他們的位置從 0 到 N-1 開始編號。騎馬比武的主持人每一回合叫出兩個位置 S 跟 E (其中 $0 \leq S < E \leq N - 1$)。所有介於 S 與 E (含) 這兩個位置的騎士則開始進行騎馬比武。最後的贏家可以留下來繼續進行競賽，並回到他原來的位置，而輸家則離開這個競賽。在這之後，剩下的騎士按照原來排列的順序，往前擠掉空出來的位置。所以他們的位置編號變成從 0 到 N - (E - S) - 1。騎馬比武競賽的主持人接著進行下一個回合的比賽，直到最後剩下唯一個騎士。

Leonardo 知道所有騎士有不同的強度，這個強度從 0 (最弱) 到 N-1 (最強)。他也知道騎馬比武競賽的主持人會下怎麼樣的命令來進行C回合的競賽，畢竟他是無所不能的 Leonardo...。而且他也確定在每一個回合中，擁有最大強度的騎士會獲得勝利。

遲到的騎士

N個騎士中的N-1個騎士已經排成了一排，只是最受歡迎的騎士還未出現。這個騎士的強度為 R 但是他遲到了。為了讓這場競賽達到最高潮, Leonardo 想要讓這個騎士能好好展現他的風采，所以想要幫他安插一個位置，而這個位置可以使得這個騎士能獲得最多回合的勝利。請注意，我們不關心與此騎士無關的回合。我們只關心包含此騎士而且由他贏得勝利的回合。

例子'

假設有5個騎士，其中四個騎士已經排列好，而他們的強度分別是[1,0,2,4]。而遲到騎士的強度為 3 。假設要進行 3 回合，騎馬比武的主持人打算要叫出的位置(S,E) 分別是 (1, 3), (0, 1), (0, 1)。

假設 Leonardo 將遲到的騎士插到第一個位置而且遲到的騎士強度為3。那麼騎士強度的排列將會是 [3, 1, 0, 2, 4]。第一回合參與的騎士為位置 1,2,3的騎士，他們的強度分別是1,0,2，所以由強度2的騎士獲得勝利。經過這一回合，新的騎士強度的排列變成 [3, 2, 4]。下一個回合是由強度3與強度2 (位置0,1)的騎士進行比賽，由強度3的騎士獲得勝利。而騎士強度的排列則變成 [3,4]。最後一回合 (位置0,1)由強度4的騎士獲得勝利。那麼，遲到的騎士只有獲得一回合的勝利 (第二回合)。

若 Leonardo 將遲到的騎士插入強度 1 與強度 0的騎士中間，騎士強度的排列將會是 [1, 3, 0, 2, 4]。這一次, 第一回合比賽的騎士強度為 3,0,2。由強度3的騎士獲勝，然後騎士強度的

排列變成 [1,3,4]。在第二回合中由強度1對上強度3的騎士，由強度3的騎士獲勝。最後的一回合，騎士強度的排列變成 [3,4]，由強度4的騎士獲得勝利。在這個排列中，遲到的騎士獲得兩回合的勝利。這實際上是最佳的位置，因為沒有其他的位置可以讓遲到的騎士獲得兩回合以上的勝利。

Statement

你的任務是寫一個程式來幫遲到的騎士選擇最佳的位置讓他能獲得最多的勝利回合數，以符合Leonardo 的期待。具體而言，你必須實作一個副程式 `GetBestPosition`(N, C, R, K, S, E), 其中:

- N 是騎士的個數 ;
- C 是競賽主持人會進行的回合數 ($1 \le C \le N - 1$);
- R 是遲到的騎士的強度 ; 所有騎士的強度都是不一樣的，並介於 0, …, N - 1 (含)。只有遲到騎士的強度 R 是特別明確給定的。
- K 是一個 N-1 個整數構成的陣列, 表示已經排列好成一列的 N-1 個騎士的強度 ;
- S 和 E 是兩個大小為C的陣列: 對每一個介於 0 與 C-1的i (含)，競賽主持人進行第 (i + 1) 回合比賽的騎士為從位置 S[i] 到位置 E[i] (含). 你可以假設對每一個 i , $S[i] < E[i]$.

呼叫此副程式的參數都是合法的; 在第 (i+1) 回合，E[i] 會小於這個回合剩下的騎士數量。經過 C回合的命令之後，只會剩下一個騎士。

`GetBestPosition`(N, C, R, K, S, E) 必須回傳最佳的位置 P 好讓 Leonardo 可以將遲到的騎士安插進去($0 \le P \le N - 1$). 如果有多於一個的位置，請輸出編號最小的位置。位置 P 是在插入遲到的騎士之後，由0開始起算到該騎士的位置。也就是在最佳解中，P是排列在遲到的騎士之前的騎士個數。具體而言，P=0 代表遲到的騎士排在最前面，而P=N-1 代表遲到的騎士排在最後面。

Subtask 1 [17 points]

假設$N \le 500$.

Subtask 2 [32 points]

假設$N \le 5{,}000$.

Subtask 3 [51 points]

假設 $N \le 100{,}000$.

實作細節

你必須上傳一個檔案叫做 `tournament.c`, `tournament.cpp` 或 `tournament.pas`. 這個檔案必須實作一個副程式，其函數原型如下所述。

C/C++ programs

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

Pascal programs

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

這些副程式必須表現得如前所述。當然你也可以實作任何自用的副程式。你所上傳的檔案不能與標準輸入/輸出或其他檔案進行互動。

範例評分系統

所提供範例評分系統會期待符合下面格式的輸入:

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

時間與記憶體限制

- 時間限制: 1 秒.
- 記憶體限制: 256 MiB.