Giải đấu thương

Để tổ chức đám cưới với Beatrice Este năm 1491, Công tước thành Milan Lodovico Sforza yêu cầu Leonardo sắp đặt cẩn thận buổi tiệc cưới, kể cả việc tổ chức giải đấu thương lớn diễn ra trong suốt ba ngày. Thế nhưng hiệp sĩ được hâm mộ nhất đã đến muộn …

Giải đấu

Trong giải đấu, N hiệp sĩ thoạt đầu được xếp thành một hàng và tiếp đến vị trí của họ được đánh số từ 0 đến N-1 theo thứ tự trong hàng. Trọng tài thiết lập một *vòng đấu* bằng việc gọi ra hai vị trí S và E (trong đó $0 \leq S < E \leq N - 1$). Tất cả các hiệp sĩ ở vị trí giữa S và E (kể cả hai đầu mút) sẽ tranh tài: người chiến thắng sẽ được tiếp tục thi đấu và sẽ quay lại vị trí của mình trong hàng, còn các hiệp sĩ thua sẽ bị loại khỏi cuộc chơi và rời khỏi vị trí. Sau đó, các hiệp sĩ còn lại được dồn về phía đầu hàng, giữ nguyên vị trí tương đối trong hàng sao cho các vị trí của họ là từ 0 đến N - (E - S) - 1. Trọng tài lại thiết lập một vòng đấu mới, lặp lại quá trình này cho đến khi chỉ còn một hiệp sĩ.

Leonardo biết rằng các hiệp sĩ có sức mạnh khác nhau thể hiện bởi thứ hạng đôi một khác nhau từ 0 (yếu nhất) đến N-1 (mạnh nhất). Ông cũng biết chính xác các khẩu lệnh mà trọng tài sẽ đưa ra để thực hiện C vòng đấu: vì ông ấy là Leonardo, và ông ấy biết chắc là trong mỗi vòng đấu hiệp sĩ có thứ hạng cao nhất sẽ giành chiến thắng.

Hiệp sĩ đến muộn

Có N - 1 trong số N hiệp sĩ đã đứng trong hàng, chỉ thiếu mặt hiệp sĩ được hâm mộ nhất. Hiệp sĩ này có thứ hạng R và đến hơi muộn một chút. Để tăng tính hấp dẫn của giải đấu, Leonardo muốn khai thác sự nổi tiếng của hiệp sĩ này và muốn chọn cho hiệp sĩ một vị trí trong hàng sao cho số lượng vòng đấu mà hiệp sĩ đến muộn giành chiến thắng là lớn nhất. Chú ý là chúng ta không quan tâm đến các vòng đấu mà hiệp sĩ đến muộn không tham gia, mà chỉ quan tâm đến những vòng đấu hiệp sĩ này tham gia và giành chiến thắng.

Ví dụ

Vói N = 5 hiệp sĩ, N - 1 hiệp sĩ đã được xếp trong hàng có thứ hạng tương ứng là [1, 0, 2, 4]. Tiếp theo, hiệp sĩ đến muộn có thứ hạng R = 3. Với C = 3 vòng đấu, trọng tài dự tính gọi cặp vị trí (S, E) cho mỗi vòng đấu theo thứ tự là: (1, 3), (0, 1), (0, 1).

Nếu Leonardo xếp hiệp sĩ đến muộn vào vị trí đầu tiên, dãy thứ hạng của các hiệp sĩ trong hàng sẽ là [3, 1, 0, 2, 4]. Vòng đấu thứ nhất sẽ liên quan đến các hiệp sĩ (ở vị trí 1, 2, 3) với thứ hạng 1, 0, 2, và kết thúc với hiệp sĩ có thứ hạng 2 là người chiến thắng. Hàng mới thu được là [3, 2, 4]. Vòng đấu tiếp theo là 3 đấu với 2 (ở vị trí 0, 1), và hiệp sĩ với thứ hạng R = 3 giành chiến thắng, dẫn đến

hàng [3, 4]. Vòng đấu cuối cùng (tại vị trí 0, 1) có 4 là người chiến thắng. Như vậy, hiệp sĩ đến muộn giành chiến thắng trong chỉ một vòng đấu (vòng đấu thứ hai).

Thế nhưng, nếu Leonardo xếp hiệp sĩ đến muộn vào giữa hai hiệp sĩ có thứ hạng 1 và 0, hàng có dạng: [1, 3, 0, 2, 4]. Theo cách xếp này, vòng đấu thứ nhất liên quan đến 3, 0, 2, và hiệp sĩ với thứ hạng R = 3 giành chiến thắng. Hàng tiếp theo là [1, 3, 4], và trong vòng đấu tiếp theo (1 đấu với 3) hiệp sĩ có thứ hạng R = 3 lại chiến thắng. Hàng cuối cùng là [3, 4], trong đó 4 giành chiến thắng. Như vậy, hiệp sĩ đến muộn giành chiến thắng trong 2 vòng đấu: trên thực tế đây là cách chọn vị trí tốt nhất có thể vì không có cách xếp để hiệp sĩ đến muộn giành nhiều hơn hai chiến thắng.

Phát biểu bài toán

Nhiệm vụ của bạn là viết chương trình lựa chọn vị trí tốt nhất cho hiệp sĩ đến muộn sao cho số lượng vòng đấu mà hiệp sĩ này giành được chiến thắng là lớn nhất, đó chính là điều mà Leonardo mong muốn. Cụ thể, bạn phải cài đặt chương trình con gọi là `GetBestPosition`(N, C, R, K, S, E), trong đó:

- N là số lượng hiệp sĩ;
- C là số lượng vòng đấu mà trọng tài sẽ ra khẩu lệnh ($1 \leq C \leq N - 1$);
- R là thứ hạng của hiệp sĩ đến muộn; thứ hạng của các hiệp sĩ (kể cả những hiệp sĩ đã đứng trong hàng lẫn hiệp sĩ đến muộn) là các số khác nhau từng đôi được chọn trong khoảng từ 0, …, N - 1, và thứ hạng R của hiệp sĩ đến muộn được cho dưới dạng hiện mặc dù thứ hạng này có thể suy diễn được;
- K là mảng gồm N - 1 số nguyên biểu diễn thứ hạng của N - 1 hiệp sĩ đã đứng sẵn trong hàng ban đầu;
- S và E là hai mảng kích thước C: với mỗi i trong khoảng giữa 0 và C - 1, kể cả hai đầu mút, vòng đấu thứ (i + 1) được trọng tài ra khẩu lệnh sẽ liên quan đến tất cả các hiệp sĩ từ vị trí S[i] tới vị trí E[i], kể cả hai đầu mút. Bạn được giả thiết là với mỗi i, S[i] < E[i].

Khẩu lệnh được truyền cho chương trình này là hợp cách: Ta có E[i] là nhỏ hơn số lượng hiện thời các hiệp sĩ còn lại cho vòng đấu thứ (i + 1), và sau tất cả C khẩu lệnh chỉ còn lại duy nhất một hiệp sĩ.

`GetBestPosition`(N, C, R, K, S, E) phải trả lại vị trí P mà tại đó Leonardo phải xếp chỗ cho hiệp sĩ đến muộn ($0 \leq P \leq N - 1$). Nếu có nhiều vị trí tương đương, *hãy đưa ra vị trí nhỏ nhất*. (Vị trí P là vị trí cơ sở 0 của hiệp sĩ đến muộn trong hàng thu được. Nói cách khác, P là số lượng hiệp sĩ đứng trước hiệp sĩ đến muộn trong lời giải tối ưu. Nói riêng, P = 0 nghĩa là hiệp sĩ đến muộn đứng ở đầu hàng và P = N - 1 nghĩa là hiệp sĩ đến muộn đứng ở cuối hàng.)

Subtask 1 [17 points]

Bạn được giả thiết là $N \leq 500$.

Subtask 2 [32 points]

Bạn được giả thiết là $N \leq 5\,000$.

Subtask 3 [51 points]

Bạn được giả thiết là $N \leq 100\,000$.

Chi tiết cài đặt

Bạn phải nộp đúng một file, được gọi là `tournament.c`, `tournament.cpp` hoặc `tournament.pas`. File này phải cài đặt chương trình con mô tả ở trên sử dụng các mẫu sau đây.

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;
```

Các chương trình này phải hoạt động như đã mô tả ở trên. Tất nhiên bạn có thể cài đặt thêm những chương trình con khác để sử dụng nội bộ. Các lần giao nộp không được giao tiếp dưới bất cứ hình thức nào với input/output chuẩn, cũng như với bất cứ file nào khác.

Sample graders

Chương trình grader được cung cấp cùng với môi trường tác nghiệp sẽ nhận đầu vào theo khuôn dạng sau:

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

Giới hạn thời gian và bộ nhớ

- Giới hạn thời gian: 1 giây.
- Giới hạn bộ nhớ: 256 MiB.