טורניר אבירים

לאונרדו מארגן עבור הדוכס טורניר קרבות אבירים. יש בממלכה אביר אחד שידוע בתור "האביר המגניב" (Late Knight) והוא מאחר לטורניר.

תיאור מהלך הטורניר

בטורניר משתתפים N אבירים. בהתחלה, הם עומדים כולם בשורה בעמדות שממוספרות מ- 0 עד N-1. יש אדם שתפקידו הוא להיות שופט הטורניר (Joust Master). הטורניר מחולק לסדרה של סיבובים. כדי להחליט מי יתחרה בסיבוב, השופט בוחר שתי עמדות בשורה שיקראו S ו- E (מתקיים $0 \le S < E \le N - 1$). כל האבירים שהעמדה שלהם בשורה היא מהמקום ה- S עד למקום ה- E (כולל S ו- E) מתחרים בסיבוב. אביר אחד מנצח בסיבוב והוא חוזר למקומו בשורה. כל האבירים האחרים שהתחרו באותו סיבוב מודחים מהטורניר ולא חוזרים אליו. לאחר-מכן, כל האבירים שנשארו בטורניר מצמצמים רווחים בשורה, כאשר הם שומרים על הסדר ביניהם. לכן, העמדות של האבירים לאחר הסיבוב ממוספרות מ- 0 עד N - (E - S) - 1. אחר-כך, מתחילים סיבוב חדש באותו אופן וממשיכים באופן זה עד שנשאר רק אביר אחד בטורניר.

לכל אביר נתונה דרגת החוזק שלו (Rank). דרגת החוזק של אביר היא מספר שלם בין 0 (החלש ביותר) ל - N-1 (החזק ביותר). לאבירים שונים יש דרגות חוזק שונות. בכל סיבוב, האביר שמנצח הוא האביר הכי חזק מבין המשתתפים בסיבוב. הערכים של S ו - E עבור כל אחד מהסיבובים ידועים מראש.

האביר המגניב

מתוך N האבירים, N-1 אבירים כבר מסודרים בשורה. האביר החסר הוא האביר המגניב שמגיע באיחור קל. דרגת החוזק שלו היא R. ליאונרדו צריך להחליט איפה בשורה להכניס את האביר המגניב. "סיבוב מגניב" הוא סיבוב שבו האביר המגניב משתתף ומנצח. ליאונרדו רוצה שיהיו בטורניר כמה שיותר סיבובים מגניבים. לשם כך, הוא צריך למקם את האביר המגניב בעמדה אופטימלית בשורה.

Example

For N = 5 knights, the N - 1 knights that are already arranged in the line have ranks [1, 0, 2, 4], respectively. Consequently, the late knight has rank R = 3. For the C = 3 rounds, the joust master intends to call out the (S, E) positions per round, in this order: (1, 3), (0, 1), (0, 1).

If Leonardo inserts the late knight at the first position, the ranks of the knights on the line will be [3, 1, 0, 2, 4]. The first round involves knights (at positions 1, 2, 3) with ranks 1, 0, 2, leaving the knight with rank 2 as the winner. The new line is [3, 2, 4]. The next round is 3 against 2 (at positions 0, 1), and knight with rank R = 3 wins, leaving the line [3, 4]. The final round (at positions 0, 1) has 4 as winner. Thus, the late knight only wins one round (the second one).

Instead, if Leonardo inserts the late knight between those two of ranks 1 and 0, the line looks like this: [1, 3, 0, 2, 4]. This time, the first round involves 3, 0, 2, and the knight with rank R = 3 wins. The next starting line is [1, 3, 4], and in the next round (1 against 3) the knight with rank R = 3

wins again. The final line is [3, 4], where 4 wins. Thus, the late knight wins two rounds: this is actually the best possible placement as there is no way for the late knight to win more than twice.

תיאור הבעיה

המשימה היא לכתוב תוכנית שמחשבת את המקום בשורה שבו כדאי לשים את האביר המגניב כדי שיהיו כמה שיותר סיבובים מגניבים. מבחינת המימוש, צריך לממש את הפונקציה `GetBestPosition(N, C, R, K, S, E)`, כאשר:

- N הוא מספר האבירים;
- C הוא מספר הסיבובים בטורניר (מתקיים $1 \le C \le N - 1$).
- R הוא דרגת החוזק של האביר המגניב;

K הוא מערך של N-1 מספרים שלמים שמתארים את דרגות החוזק של האבירים (חוץ מהאביר המגניב) לפי הסדר שלהם בשורה (המערך K מכיל את כל המספרים בין 0 ל - N-1, פרט למספר R).

- S ו - E הם מערכים של מספרים שלמים שבכל אחד מהם C איברים. מערכים אלה מגדירים את הטווחים שבוחר השופט בכל סיבוב. לכל i מ- 0 עד C-1 (כולל), מתקיים שבסיבוב ה- i+1 בטורניר יתחרו האבירים שעומדים בתחילת הסיבוב בעמדות מ- S[i] עד ל- E[i] (כולל). לכל i, מתקיים $S[i] < E[i]$.

הנתונים הגיוניים: מובטח שלכל i, המספר E[i] קטן ממספר האבירים ששרדו עד לתחילת הסיבוב ה- i+1. בנוסף, מובטח שלאחר C סיבובים ישאר בדיוק אביר אחד בטורניר.

על הפונקציה `GetBestPosition(N, C, R, K, S, E)` להחזיר את העמדה P שבה לאונרדו צריך לשים את האביר המגניב ($0 \le P \le N - 1$). אם יש מספר עמדות אפשריות *עליכם להחזיר את העמדה המינימלית מביניהן* . (הבהרה: P הוא מספר האבירים שיעמדו בשורה לפני האביר המגניב. בפרט, אם $P = 0$ אז האביר המגניב עומד בתחילת השורה. אם $P = N-1$ אז האביר המגניב עומד בסוף השורה.)

[Subtask 1 [17 points

You may assume that $N \le 500$.

[Subtask 2 [32 points

You may assume that $N \le 5\,000$.

[Subtask 3 [51 points

You may assume that $N \le 100\,000$.

Implementation details

You have to submit exactly one file, called `tournament.c`, `tournament.cpp` or `tournament.pas`. This file must implement the subprogram described above using the .following signatures

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;
```

These subprograms must behave as described above. Of course you are free to implement other subprograms for their internal use. Your submissions must not interact in any way with standard .input/output, nor with any other file

Sample graders

:The sample grader provided with the task environment will expect input in the following format

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

Time and Memory limits

- .Time limit: 1 second
- .Memory limit: 256 MiB