

## Hasta Siempre, Comandante Chen Guevara! (hasta)

Gli alimenti scarseggiano, e il popolo è costretto a fare lunghe file per i viveri. Il compagno Jacopo vuole cogliere l'occasione per fare scoppiare la rivoluzione. Nel paese ci sono  $K$  classi sociali. In fila ci sono  $N$  persone, ed ognuna di esse è identificata dalla sua classe sociale. Jacopo, grazie ai suoi superpoteri, è in grado di sobillare un certo segmento  $[l, r]$  (con  $0 \leq l \leq r < N$ ) della fila, ovvero tutti i cittadini in posizione  $i$  tale che  $l \leq i \leq r$  sentiranno il desiderio di rivoluzione. Jacopo ha notato che non sempre le rivoluzioni vanno a buon fine, ma forte della matematica, per ogni diversa classe sociale  $i$  ha calcolato il valore  $R_i$ , ovvero il numero perfetto di cittadini appartenenti a quella classe sociale per la rivoluzione. Infatti se fossero meno non avrebbero le forze per la rivoluzione, e se fossero di più finirebbero per mettersi a litigare. Jacopo, dopo averti fornito una descrizione della fila ed il numero rivoluzionario di ogni classe sociale, ti fa diverse domande, chiedendoti "se istigassi il segmento  $l, r$ , quanti si unirebbero alla rivoluzione?", ovvero, quante persone appartengono alle classi sociali tali che il segmento  $[l, r]$  contiene esattamente  $R_i$  cittadini appartenenti all' $i$ -esima classe sociale?

### Implementazione

Dovrai sottoporre esattamente un file con estensione `.c`, `.cpp`

📎 Tra gli allegati a questo task troverai un template (`hasta.c`, `hasta.cpp` con un esempio di implementazione.

Dovrai implementare le seguenti funzione:

#### ■ Funzione inizia

```
C/C++ void inizia(int N, int K, int V[], int R[]);
```

- L'intero  $N$  rappresenta il numero di cittadini in fila.
- L'intero  $K$  rappresenta il numero di classi sociali.
- L'array  $V$ , indicizzato da 0 a  $N - 1$ , è tale per cui  $V[i]$  indica la classe sociale del  $i$ -esimo cittadino
- L'array  $R$ , indicizzato da 0 a  $K - 1$ , è tale per cui  $R[i]$  indica il valore di rivoluzione dell' $i$ -esima classe sociale

#### ■ Funzione istiga

```
C/C++ int istiga(int l, int r);
```

- Gli interi  $l$  ed  $r$  indicano il segmento che interessa a Jacopo.
- La funzione deve restituire il numero di persone che lo seguirebbero nella rivoluzione.

Il grader chiamerà una sola volta la funzione `inizia` all'inizio; dopodiché chiamerà più volte la funzione `istiga`, e stamperà i risultati delle query sul file di output.

### Grader di prova

Nella directory relativa a questo problema è presente una versione semplificata del grader usato durante la correzione, che potete usare per testare le vostre soluzioni in locale. Il grader di esempio legge i dati di input dal file `stdin`, chiama le funzioni che dovete implementare e scrive il file `stdout`, secondo il seguente formato.

Il file `stdin` è composto da  $Q + 4$  righe, contenenti:

- Riga 1: i due interi  $N$  e  $K$ .
- Riga 2: gli interi  $V_0, \dots, V_{N-1}$
- Riga 3: gli interi  $R_0, \dots, V_{K-1}$
- Riga 4: l'unico intero  $Q$ , il numero di query.
- Righe 5,  $\dots$ ,  $Q + 4$ : le query, nel formato  $l\ r$ .

Il file `stdout` è composto da un'unica riga, contenente:

- Riga 1: i  $Q$  interi di risposta alle query, ritornati dalle chiamate a `istiga`.

## Assunzioni

- $1 \leq K \leq N \leq 200\,000$ .
- $0 \leq V_i < K$ .
- $0 \leq R_i < 10^9$ .
- In ogni chiamata ad `istiga`  $0 \leq l \leq r < N$
- `istiga` viene chiamata al più 200 000 volte.

## Assegnazione del punteggio

Il tuo programma verrà testato su diversi test case raggruppati in subtask. Per ottenere il punteggio relativo ad un subtask, è necessario risolvere correttamente tutti i test relativi ad esso.

- **Subtask 1** [0 punti]: Casi d'esempio.
- **Subtask 2** [20 punti]:  $N \leq 2000$ , e `istiga` viene chiamata al più 2000 volte.
- **Subtask 3** [20 punti]:  $K \leq 100$ , e `istiga` viene chiamata al più 70 000 volte.
- **Subtask 4** [20 punti]:  $r = N - 1$
- **Subtask 5** [20 punti]: `istiga` viene chiamata con  $r$  crescente.
- **Subtask 6** [20 punti]: Nessuna limitazione specifica.

## Esempi di input/output

| input                                      | output           |
|--|------------------|
| <pre> 5 2 0 1 0 0 0 2 3 1 0 2 </pre>       | <pre> 2 </pre>   |
| <pre> 5 3 0 0 1 1 2 1 2 1 2 0 0 0 4 </pre> | <pre> 1 3 </pre> |

## Spiegazione

Nel **primo caso di esempio**, Jacopo verrà seguito dai cittadini 0 e 2, appartenenti alla classe sociale 0.

Nel **secondo caso di esempio**, la prima volta Jacopo viene seguito dal cittadino 0 della classe sociale 0, mentre la seconda volta verrà seguito dai cittadini 2 e 3 della classe sociale 1, e dal cittadino 4 della classe sociale 2.