

Episodio III: la coda per il buffet (coda)

Il fatidico giorno della gara è arrivato! Per la prima volta nella storia, italiani e alieni insieme competono sulle loro abilità informatiche. Purtroppo, dopo pochi minuti di gara, grazie ad un sofisticato algoritmo di intelligenza artificiale, lo staff si accorge che la gara di quest'anno è troppo facile.

Secondo l'algoritmo, tutti gli N partecipanti otterranno un punteggio pieno: più precisamente, il partecipante i -esimo ($i = 0, \dots, N - 1$) otterrà punteggio pieno dopo H_i secondi dall'inizio, lasciando quindi l'area di gara per raggiungere immediatamente il buffet.

Come è noto, i partecipanti delle OII sono piuttosto prevedibili. Se arrivano al buffet quando non è ancora aperto, oppure trovano già K persone in coda per il buffet, se ne vanno subito verso il paninaro più vicino. Altrimenti si accodano, aspettando pazientemente il proprio turno, mentre i camerieri servono una persona in coda ogni secondo. Dopo X secondi da inizio gara, il buffet viene chiuso e tutti i partecipanti in coda sono costretti a ripiegare sul paninaro. Aiuta lo staff a pianificare il buffet, calcolando quante persone vengono servite per ogni possibile orario di apertura!



Figura 1: Il lauto buffet che vi aspetta.

Implementazione

Dovrai sottoporre un unico file, con estensione `.cpp`.

👉 Tra gli allegati a questo task troverai un template `coda.cpp` con un esempio di implementazione.

Dovrai implementare la seguente funzione:

```
C++ | vector<int> cucina(int N, int K, int X, vector<int> H);
```

- L'intero N rappresenta il numero di partecipanti della gara.
- L'intero K rappresenta la lunghezza massima della coda ammessa dai partecipanti.
- L'intero X rappresenta il numero di secondi da inizio gara della chiusura del buffet.
- Il vettore H , indicizzato da 0 a $N - 1$, contiene il secondo a cui ciascun partecipante arriva al buffet.
- La funzione dovrà restituire un vettore R di lunghezza X , contenente il numero di persone servite per ogni orario di apertura possibile.

Il grader chiamerà la funzione `cucina` e ne stamperà i valori restituiti 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 da `stdin`, chiama le funzioni che dovete implementare e scrive su `stdout`, secondo il seguente formato.

Il file di input è composto da due righe, contenenti:

- Riga 1: gli interi N , K e X .
- Riga 2: gli interi H_0, H_1, \dots, H_{N-1} .

Il file di output è composto da un'unica riga contenente gli X interi C_0, C_1, \dots, C_{X-1} restituiti dalla funzione cucina.

Assunzioni

- $1 \leq N, K, X \leq 500\,000$.
- $0 \leq H_i < X$ per ogni $0 \leq i \leq N - 1$.

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 che lo compongono.

- **Subtask 1 [0 punti]**: Casi d'esempio.
- **Subtask 2 [12 punti]**: $N, X \leq 1000$.
- **Subtask 3 [21 punti]**: $N \leq 1000$.
- **Subtask 4 [13 punti]**: $K = N$.
- **Subtask 5 [23 punti]**: $K \leq 10$.
- **Subtask 6 [31 punti]**: Nessuna limitazione aggiuntiva.

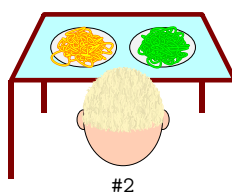
Esempi di input/output

stdin	stdout
6 2 4 3 3 0 2 3 2	3 2 2 1
7 3 6 4 0 4 3 0 0 2	6 4 4 3 2 0

Spiegazione

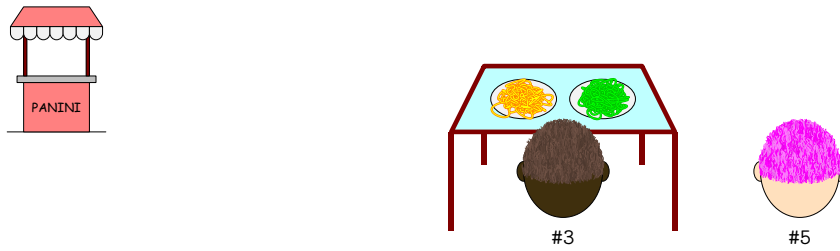
Nel **primo caso di esempio**, aprendo il buffet dall'inizio della gara:

- Al secondo 0 il partecipante 2 arriva al buffet, si mette in coda e viene immediatamente servito.

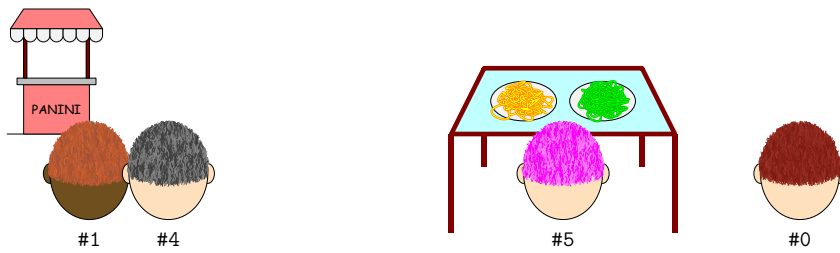


- Al secondo 1 non arriva nessun partecipante.

- Al secondo 2 arrivano due partecipanti al buffet. Entrambi si mettono in fila e il primo della coda viene servito.



- Al secondo 3 arrivano tre partecipanti al buffet. Uno di loro si mette in coda mentre gli altri se ne vanno dato che la coda è già lunga $K = 2$.

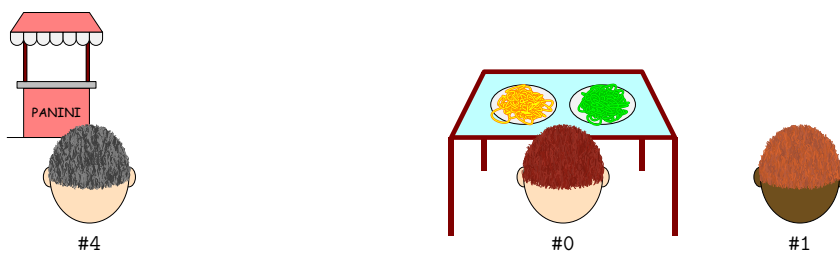


- Al secondo 4 il buffet viene chiuso e i partecipanti ancora in fila vanno al paninaro.



Aprendo il buffet al secondo 0 vengono quindi serviti 3 partecipanti. Aprendolo invece al secondo 3:

- Al secondo 0 il partecipante 2 arriva al buffet, lo trova chiuso e va al paninaro.
- Al secondo 2 arrivano due partecipanti al buffet, e trovandolo chiuso entrambi vanno al paninaro.
- Al secondo 3 arrivano tre partecipanti al buffet. Due di loro si mettono in coda mentre l'altro se ne va dato che la coda è già lunga $K = 2$. Il primo partecipante della coda viene servito.



- Al secondo 4 il buffet viene chiuso e i partecipanti in fila vanno al paninaro.



Aprendo il buffet al secondo 3 viene quindi servito 1 partecipante.