



Incontri inaspettati II (incontri2)

Gemma è in vacanza nella città turistica di Vimercate. La città offre ai visitatori N attrazioni turistiche, ciascuna delle quali ha un valore di bellezza A_i , ogni turista che la visiterà farà un giro turistico della durata di A_i secondi. Gemma vuole visitare tutte le attrazioni partendo dall'attrazione 0 in ordine fino all'attrazione $N - 1$.

Gemma è, però, venuta a conoscenza che anche il suo stalker Emil è in città. Emil per non destare sospetti ha deciso di visitare la città come un perfetto turista partendo dall'attrazione $N - 1$ e visitando ogni attrazione in ordine inverso fino all'attrazione 0.

Sia Gemma che Emil vanno molto di fretta e non si fermeranno neanche un secondo tra un'attrazione e quella successiva (grazie anche al servizio di bus estremamente efficiente di Vimercate).

L'obiettivo di Emil è incontrare casualmente Gemma all'entrata di una meta turistica in modo da far parte dello stesso giro turistico.

Stabilisci se Emil riuscirà ad incontrare Gemma!



Il famoso Ponte San Rocco di Vimercate

Implementazione

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

📎 Tra gli allegati a questo task troverai un template (`incontri2.c`, `incontri2.cpp`) con un esempio di implementazione.

Dovrai implementare la seguente funzione:

■ Funzione visita

C/C++ `bool visita(int N, int A[]);`

La funzione viene chiamata all'inizio dell'esecuzione del programma con i seguenti parametri:

- L'intero N rappresenta il numero attrazioni turistiche di Vimercate.
- L'array A indicizzato da 0 a $N - 1$ rappresenta la bellezza di ogni attrazione.
- La funzione dovrà ritornare `true` se Emil incontrerà Gemma, `false` altrimenti.
- La funzione verrà chiamata dal grader al più **10 volte**, pertanto si consiglia di prestare attenzione alle variabili dichiarate globalmente.

Grader di prova

Tra gli allegati di 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, chiama la funzione che devi implementare e stampa l'output secondo il seguente formato.

L'input è composto da 2 righe, contenenti:

- La prima riga contiene l'intero N ,
- La seconda riga contiene N interi A_0, A_1, \dots, A_{N-1} .

L'output è composto da una sola riga contenente "Emil" se il valore di ritorno della funzione `visita` è `true`, "Gemma" altrimenti.

Assunzioni

- $2 \leq N \leq 10^5$.
- $1 \leq A_i \leq 10^4$ per ogni $0 \leq i \leq N - 1$.
- Gemma ed Emil partono allo stesso istante!
- La funzione `visita` verrà chiamata dal grader al più **10 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 [30 punti]**: $A_i = 1$ per ogni $0 \leq i \leq N - 1$.
- **Subtask 3 [30 punti]**: $N \leq 1000$.
- **Subtask 4 [40 punti]**: Nessuna limitazione specifica.

Esempi di input/output

4 2 1 3 2	Gemma
4 4 1 3 1	Emil

Spiegazioni

Nel primo caso di esempio Gemma ed Emil si troveranno entrambi all'attrazione 2 dopo 3 secondi, tuttavia fanno parte di due giri turistici diversi pertanto non si incontreranno.

Nel secondo caso di esempio sia Gemma che Emil raggiungono l'attrazione 1 dopo 4 secondi.