Treni (`treni`)

Gestire la logistica per una gara importante non è mai semplice. Quando a dover viaggiare sono persone come Ugo il compito diventa ancora più complicato: Ugo infatti ama viaggiare in treno, e farebbe di tutto per far durare il viaggio il più a lungo possibile.

Ugo deve viaggiare per N giorni, e ogni giorno può scegliere se prendere un treno della SuperFastTravels$^{(TM)}$ o della HyperFastTravels$^{(C)}$, o se non viaggiare affatto. A causa delle rivalità tra le due aziende però è possibile prendere un treno della HyperFastTravels$^{(C)}$ solo se il giorno precedente non si è preso nessun treno (è quindi possibile prenderlo anche il primo giorno).

Per ogni giorno ti viene fornita la durata della tratta offerta da entrambe le società, in decasecondi. Aiuta Ugo a calcolare quanto tempo al massimo potrà viaggiare sui suoi amatissimi treni!

Implementazione

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

> ☞ Tra gli allegati a questo task troverai un template (`treni.cpp`) con un esempio di implementazione.

Dovrai implementare la seguente funzione:

Funzione `tempo_massimo`

C++	`int tempo_massimo(int N, int A[], int B[]);`

- L'intero N indica il numero di giorni di viaggio.
- L'array `A`, indicizzato da 0 a $N-1$, rappresenta il tempo di percorrenza dei treni della SuperFastTravels$^{(TM)}$.
- L'array `B`, indicizzato da 0 a $N-1$, rappresenta il tempo di percorrenza dei treni della HyperFastTravels$^{(C)}$.

Il grader chiamerà la funzione `tempo_massimo`, che dovrà ritornare il massimo numero di decasecondi che Ugo riuscià a passare in treno durante gli N giorni.

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 da stdin, chiama la funzione che dovete implementare e scrive in stdout, secondo il seguente formato.

Il file `stdin` è composto da $N+1$ righe, contenenti:

- Riga 1: L'intero N.
- Righe $2 \dots N+1$: Due interi A_i e B_i per ogni riga rappresentanti il tempo di percorrenza, rispettivamente, della SuperFastTravels$^{(TM)}$ e della HyperFastTravels$^{(C)}$.

Il file `stdout` è composto da 1 riga: la risposta al problema.

Assunzioni

- $1 \leq N \leq 10^6$.
- $1 \leq A_i, B_i \leq 10^3$.

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 [5 punti]**: $N \leq 3$.
- **Subtask 3 [10 punti]**: $N \leq 18$.
- **Subtask 4 [30 punti]**: $N \leq 2\,000$.
- **Subtask 5 [55 punti]**: Nessuna limitazione specifica.

Esempi di input/output

3 2 5 3 2 2 8	13
4 3 2 3 3 3 1 3 4	12

Spiegazione

Nel primo caso di esempio conviene prendere un treno della HyperFastTravels$^{(C)}$ il primo e l'ultimo giorno, rimanendo fermi il secondo.

Nel secondo caso di esempio la scelta migliore è viaggiare sempre con SuperFastTravels$^{(TM)}$