

Pausa caffè (caffè)

Limite di tempo: 1.0 secondi
Limite di memoria: 256 MiB

Durante i lunghi ed estenuanti stage di preparazione dei PO (probabili olimpici), gli N tutor ogni tanto recuperano le energie andando a prendersi un caffè. Per ottimizzare la loro presenza in aula, ogni tutor fa pausa in un momento diverso del giorno: in particolare, il tutor i -esimo fa pausa nell'intervallo di tempo $[A_i, B_i]$ dove i valori A_i e B_i sono tutti distinti tra loro.

I tutor tra di loro sono sempre molto cordiali: ogni volta che uno di loro raggiunge la macchinetta del caffè, offre sempre un giro di caffè a tutti i presenti. Questo elevato consumo di caffè sta facendo rapidamente finire le scorte a disposizione della macchinetta. Aiuta Monica a programmare i rifornimenti calcolando quanti caffè vengono bevuti ogni giorno!

Implementazione

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

📁 Tra gli allegati a questo task troverai un template (`caffè.c`, `caffè.cpp`, `caffè.pas`) con un esempio di implementazione da completare.

Se sceglierai di utilizzare il template, dovrai implementare la seguente funzione:

C/C++	<code>int pausa(int N, int A[], int B[]);</code>
Pascal	<code>function pausa(N: longint; var A, B: array of longint): longint;</code>

In cui:

- L'intero N rappresenta il numero di tutor.
- Gli array A e B , indicizzati da 0 a $N - 1$, descrivono gli intervalli $[A_i, B_i]$ di tempo in cui i tutor vanno a prendersi il caffè.
- La funzione dovrà restituire il numero totale di caffè consumati, che verrà stampato sul file di output.

Dati di input

Il file `input.txt` è composto da $N + 1$ righe. La prima riga contiene l'unico intero N . Le successive N righe contengono ciascuna due interi A_i, B_i separati da uno spazio.

Dati di output

Il file `output.txt` è composto da un'unica riga contenente un unico intero, la risposta a questo problema.

Assunzioni

- $1 \leq N \leq 40\,000$.
- $0 \leq A_i < B_i \leq 1\,000\,000$ per ogni $i = 0 \dots N - 1$.
- I valori A_i e B_i sono tutti differenti tra loro.

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 [10 punti]:** Casi d'esempio.
- **Subtask 2 [20 punti]:** $N \leq 10$.
- **Subtask 3 [40 punti]:** $N \leq 1000$.
- **Subtask 4 [30 punti]:** Nessuna limitazione specifica.

Esempi di input/output

input.txt	output.txt
3 4 7 5 10 1 6	6
6 1 6 4 8 7 9 14 17 10 12 13 20	9

Spiegazione

Nel **primo caso di esempio**, prima arriva il tutor 2 che si prende il suo caffè, poi il tutor 0 che lo offre al tutor 2 e infine il tutor 1 che lo offre a entrambi gli altri per un totale di $1 + 2 + 3 = 6$ caffè.

Nel **secondo caso di esempio**, il tutor 1 offre il caffè al tutor 0, il tutor 2 lo offre al tutor 1, il tutor 3 lo offre al tutor 5 per un totale di $1 + 2 + 2 + 2 + 1 + 1 = 9$ caffè.