Parco divertimenti (parco)

Elia ha deciso di visitare il nuovo parco divertimenti di Cesena. Per sfruttare al meglio il suo tempo, si è posto l'obiettivo di divertirsi il più possibile prima dell'orario di chiusura **senza mai prendere la stessa giostra più di una volta**, altrimenti si annoierebbe.

Da bravo informatico, prima di comprare il biglietto, Elia ha eseguito un sopralluogo per capire la struttura del parco e valutare quali attrazioni sembrano più divertenti.

Figura 1: Elia mentre esegue un meticoloso sopralluogo.

Il parco è composto da N aree diverse, ognuna con un tema particolare. Ogni area contiene K_i attrazioni, ognuna con una sua durata $W_{i,j}$ e un divertimento associato $F_{i,j}$. Il **divertimento totale** della visita di Elia sarà la somma dei divertimenti associati alle attrazioni che visiterà.

Tuttavia, il parco divertimenti è molto affollato: ogni volta che si vuole entrare in un'area bisogna fare una coda di X minuti.

Aiuta Elia a capire quanto potrà divertirsi al massimo, sapendo che ha ancora T minuti a disposizione prima della chiusura del parco.

Implementazione

Dovrai sottoporre un unico file, con estensione `.cpp`, `.py`, `.cs` o `.java`.

> ⇨ Tra gli allegati di questo task troverai dei template `parco.*` con un esempio di implementazione.

Dovrai implementare la seguente funzione:

C++	`long long int visita(int N, int X, int T, vector<int> K, vector<vector<int>> W, vector<vector<int>> F);`
Python	`def visita(N: int, X: int, T: int, K: list[int], W: list[list[int]], F: list[list[int]]) -> int:`

Java	`public static long visita(int N, int X, int T, int[] K, int[][] W, int[][] F)`
C#	`public static long visita(int N, int X, int T, int[] K, int[][] W, int[][] F)`

- L'intero N rappresenta il numero di aree.
- L'intero X rappresenta il tempo necessario per entrare in un'area.
- L'intero T rappresenta il tempo a disposizione.
- L'array K, indicizzato da 0 a $N-1$, contiene il numero di attrazioni nell'i-esima area.
- L'array W, indicizzato da 0 a $N-1$, contiene la durata delle attrazioni nell'i-esima area. In particolare per ogni $0 \leq i < N$, W_i è un array indicizzato da 0 a $K_i - 1$.
- L'array F, indicizzato da 0 a $N-1$, contiene il divertimento delle attrazioni nell'i-esima area. In particolare per ogni $0 \leq i < N$, F_i è un array indicizzato da 0 a $K_i - 1$.
- La funzione deve restituire un intero, il massimo divertimento totale che Elia può ottenere.

Grader di prova

Nella directory relativa a questo problema è presente una versione semplificata del grader usato durante la correzione, che puoi usare per testare le tue soluzioni in locale. Il grader di esempio legge i dati da `stdin`, chiama la funzione che devi implementare e scrive su `stdout`, secondo il seguente formato.

Il file di input è composto da $N+1$ blocchi, contenenti:

- Blocco 1: gli interi N, X e T.
- Blocco $1+i$:
  - Riga 1: L'intero K_i.
  - Riga $1+j$: gli interi $W_{i,j}$ e $F_{i,j}$ per $0 \leq j < K_i$.

Il file di output è composto da un'unico valore, il valore restituito dalla funzione `visita`.

Assunzioni

- $1 \leq N \leq 10\,000$.
- $1 \leq X \leq 100$.
- $1 \leq T \leq 50\,000$.
- $K_0 + K_1 + ... + K_{N-1} \leq 10\,000$.
- $0 \leq W_{i,j} \leq 100$ per ogni $0 \leq i < N$, $0 \leq j < K_i$.
- $0 \leq F_{i,j} \leq 1\,000\,000\,000$ per ogni $0 \leq i < N$, $0 \leq j < K_i$.

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]**: $K_0 + K_1 + ... + K_{N-1} \leq 20$.
- **Subtask 3 [14 punti]**: $X = 0, W_{i,j} = 1$.
- **Subtask 4 [23 punti]**: $X = 0$.
- **Subtask 5 [17 punti]**: $N \leq 5$, $T \leq 10\,000$.
- **Subtask 6 [34 punti]**: Nessuna limitazione aggiuntiva.

Esempi di input/output

stdin	stdout
2 5 15 3 3 100 4 150 5 200 4 5 100 7 100 6 200 4 50	350
3 2 20 2 5 100 7 150 2 6 200 7 300 1 5 600	900

Spiegazione

Nel **primo caso d'esempio** Elia ha 15 minuti per visitare le due aree che hanno code da 5 minuti. La soluzione ottimale è fare 5 minuti di coda per entrare nella prima area, dunque visitare la seconda e la terza attrazione impiegando $4 + 5$ minuti. Il divertimento totale sarà $150 + 200 = 350$ in 14 minuti.

Nel **secondo caso d'esempio** Elia ha 20 minuti per visitare le tree aree che hanno code da 2 minuti. La soluzione ottimale è fare 2 minuti di coda per entrare nella seconda area e visitare la seconda attrazione impiegando 7 minuti, quindi fare 2 minuti di coda per entrare nella terza area e visitare la prima attrazione in 5 minuti. Il divertimento totale sarà $300 + 600 = 900$ in $2 + 7 + 2 + 5 = 16$ minuti.