

Tris in solitaria 2 (solitario2)

Gabriele, sempre più ossessionato dal gioco del tris, ha concluso il suo precedente percorso di allenamento senza riuscire a vincere contro Giorgio.

Ha quindi ideato un nuovo programma di allenamento: presa una griglia composta da $N \times M$ caselle, vuole riempirla con il massimo numero possibile di X senza formare tris, ma alcune celle sono già state riempite con una X prima di iniziare.

È garantito che prima che Gabriele inizi il suo allenamento sulla griglia non c'è nessun tris. Quante nuove X può inserire al massimo senza formare nessun tris?

Implementazione

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

📖 Tra gli allegati a questo task troverai un template (`solitario2.c`, `solitario2.cpp`) con un esempio di implementazione da completare.

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

C	<code>int gioca(int N, int M, int** G);</code>
C++	<code>int gioca(int N, int M, vector<vector<int>> G);</code>

In cui:

- Gli interi N , M rappresentano le dimensioni della griglia.
- L'array bidimensionale G , indicizzato da 0 a $N - 1$ e da 0 a $M - 1$, contiene lo stato iniziale della griglia. Uno 0 rappresenta una cella vuota, un 1 una cella occupata da una X.
- La funzione dovrà restituire il numero massimo di X che si possono aggiungere senza creare tris, che verrà stampato sul file di output.

Dati di input

L'input è composto da $N + 1$ righe:

- La prima un'unica riga contenente i due interi N ed M .
- Le successive N righe, ognuna contenente M interi separati da spazio, descrivono lo stato iniziale della griglia. Uno 0 rappresenta una cella vuota, un 1 una cella occupata.

Dati di output

L'output è composto da un'unica riga contenente un unico intero, la risposta al problema.

Assunzioni

- $1 \leq N, M \leq 10$.
- $N \times M \leq 42$.
- È garantito che lo stato iniziale della griglia non contiene nessun tris.

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** [10 punti]: $N \leq 2$.
- **Subtask 3** [20 punti]: $N \times M \leq 16$.
- **Subtask 4** [20 punti]: $N \times M \leq 25$.
- **Subtask 5** [10 punti]: $N \times M \leq 36$ e la griglia iniziale non ha nessuna X
- **Subtask 6** [20 punti]: $N \times M \leq 36$.
- **Subtask 7** [20 punti]: Nessuna limitazione specifica.

Esempi di input/output

stdin	stdout
2 6 0 1 0 0 0 0 0 0 0 1 0 0	6
3 3 0 0 1 0 0 0 0 0 0	5

Spiegazione

Nel **primo caso di esempio**, una possibile soluzione è la seguente, dove le X aggiunte da Gabriele sono segnate in rosso.

	×	×		×	×
×	×		×	×	

Nel **secondo caso di esempio**, una possibile configurazione massima è:

	×	×
×		×
×	×	