Pulizie d'autunno (`trendytrash`)

Edoardo ha una stanza rettangolare piena di sacchi della spazzatura, di cui si vorrebbe liberare effettuando alcuni viaggi al centro di raccolta differenziata.

La stanza si può vedere come una griglia $N \times M$ (cioè con N righe, numerate da 0 a $N-1$, ed M colonne, numerate da 0 a $M-1$). Per ogni $0 \leq i < N, 0 \leq j < M$, vi è un sacco della spazzatura all'incrocio della riga i e colonna j (per un totale di $N \cdot M$ sacchi).

Per raccogliere i sacchi, Edoardo usa un carrello con il quale percorre una riga o una colonna della stanza, raccogliendo **tutti e soli** i sacchi presenti in quella riga o colonna tra quelli rimasti. Dopodiché porta tali sacchi al centro di riciclo, depositandoli nel contenitore opportuno.

Essendosi allenato duramente in palestra durante gli ultimi mesi, Edoardo può portare quanti sacchi vuole in un singolo viaggio. Tuttavia, dato che alcuni dei sacchi contengono carta e altri plastica, non vuole mai raccogliere sacchi di due tipi diversi insieme, altrimenti sarebbe troppo complicato poi separarli al centro di riciclo.

Edoardo vuole liberarsi del maggior numero possibile di sacchi della spazzatura. Aiutalo a determinare il minimo numero di sacchi che dovranno rimanere nella stanza al termine dei suoi viaggi.

Implementazione

Dovrai sottoporre un unico file, con estensione `.cpp`.

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

Dovrai implementare la seguente funzione:

C++
```cpp
int pulisci(int N, int M, vector<string> S);
```

- Gli interi N ed M rappresentano la dimensione della stanza (numero di righe e numero di colonne).
- L'array di stringhe S, indicizzato da 0 a $N-1$, contiene il tipo dei sacchi. In particolare, per ogni $0 \leq i < N$, $0 \leq j < M$, $S[i][j]$ è il **carattere** 0 se il sacco nella posizione (i, j) contiene carta, ed è il carattere 1 se tale sacco contiene plastica.
- La funzione dovrà ritornare il minimo numero di sacchi che rimangono nella stanza se Edoardo effettua i viaggi nel modo ottimale.

Il grader chiamerà prima la funzione `pulisci` e ne stamperà il valore restituito sul file di output.

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

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

- Riga 1: gli interi N ed M.
- Riga $2+i$ ($0 \leq i < N$): la stringa $S[i]$ (ovvero i caratteri $S[i][0], \ldots, S[i][M-1]$, **senza spazi**).

Il file di output è composto da un'unica riga, contenente il valore restituito dalla funzione `pulisci`.

Assunzioni

- $2 \leq N, M \leq 2000$.
- $S[i][j] =$ `'0'` o $S[i][j] =$ `'1'` per ogni $0 \leq i < N$, $0 \leq j < M$.

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.

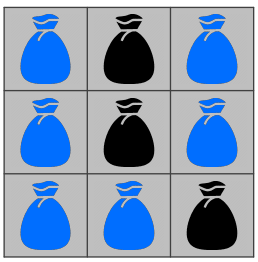
- **Subtask 1 [0 punti]**: Casi d'esempio.
- **Subtask 2 [7 punti]**: $N = 2$, $M = 2$.
- **Subtask 3 [11 punti]**: $N = 2$.
- **Subtask 4 [16 punti]**: $N \leq 5$, $M \leq 5$.
- **Subtask 5 [24 punti]**: $N \leq 100$, $M \leq 100$.
- **Subtask 6 [42 punti]**: Nessuna limitazione aggiuntiva.

Esempi di input/output

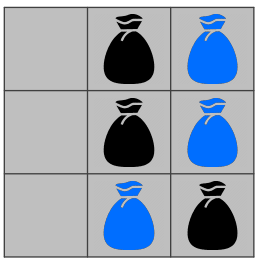
stdin	stdout
3 3 101 101 110	6
3 3 100 100 111	0
4 5 00010 01110 01000 11110	4

Spiegazione

Nel **primo caso di esempio** Edoardo può rimuovere tutti i sacchi nella colonna più a sinistra (che contengono plastica, in blu), dopodiché non potrà più portare alcun sacco, rimanendo con un totale di 6.



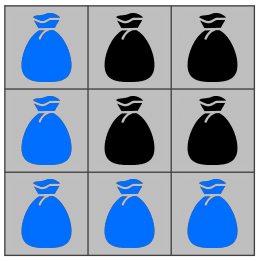
(a) Stanza iniziale



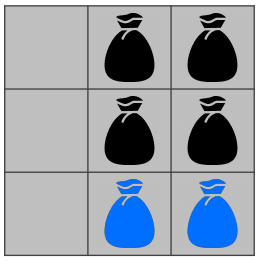
(b) Dopo il primo (e unico) viaggio

Nel **secondo caso di esempio** Edoardo può rimuovere tutti i sacchi in questo modo:

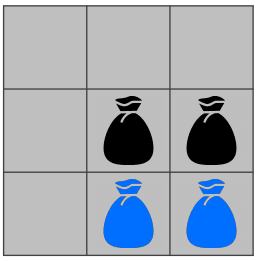
- nel primo viaggio rimuove tutti i sacchi nella colonna 0 (quella più a sinistra), che contengono plastica (in blu);
- nel secondo viaggio rimuove i due sacchi rimasti nella riga 0, che contengono carta (in nero);
- nel terzo viaggio rimuove i due sacchi rimasti nella riga 1, che contengono carta;
- nel quarto e ultimo viaggio rimuove i due sacchi rimasti nella riga 2, che contengono plastica.



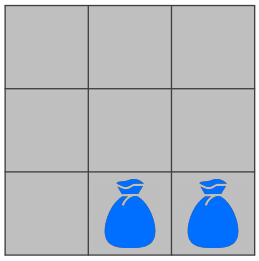
(a) Stanza iniziale



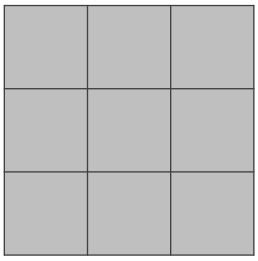
(b) Dopo il primo viaggio



(c) Dopo il secondo viaggio



(d) Dopo il terzo viaggio



(e) Dopo il quarto (e ultimo) viaggio