



Pulizie d'autunno (trendytrash)

Edoardo has a rectangular room full of trash bags and he would like to get rid of them with some trips to the recycling center.

The room can be seen as an $N \times M$ grid; with N rows, numbered from 0 to $N - 1$, and M columns, numbered from 0 to $M - 1$. For each $0 \leq i \leq N - 1, 0 \leq j \leq M - 1$, there is a trash bag at the intersection of the row i and the column j (for a total of $N \times M$ bags).

Edoardo is going to use a cart to collect the bags by going across a row or a column of the room, and collecting **all** the remaining bags in that row or column. Thereafter, he brings those bag to the recycling center, dumping them in the appropriate container.

Since he worked out hard at the gym in the past months, Edoardo can carry as many bags as he wants in a single trip. However, since some bags contain paper and others contain plastic, he doesn't want to collect bags of different types together, otherwise it would be too complex to sort them in the recycling center.

Edoardo wants to get rid of the greatest possible amount of trash bags. Help him figure out the minimum number of bags that will remain in his room after all the trips.

Implementation

You should submit a single file, with a `.cpp` extension.

📎 Among the attachments in this task you will find a template `trendytrash.cpp` with a sample implementation.

You will have to implement the following function:

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

- Integers N and M are the dimensions of the room (the number of rows and the number of columns).
- The string array S , indexed from 0 to $N - 1$, contains the types of the bags. In particular, for each $0 \leq i < N, 0 \leq j < M$, $S[i][j]$ is the **character** '0' if the bag in the position (i, j) contains paper, while it is the character '1' if the bag contains plastic.
- The function must return the minimum number of bags that remain in the room if Edoardo plans his trips optimally.

The grader will call function `pulisci` and print the return value on the output file.

Sample grader

Among this task's attachments you will find a simplified version of the grader used during evaluation, which you can use to test your solutions locally. The sample grader reads data from `stdin`, calls the function that you should implement and writes back on `stdout` using the following format.

The input file is made up of $N + 1$ lines, containing::

- Line 1: integers N and M .
- Line $2 + i$ ($0 \leq i < M$): string $S[i]$ (that is, characters $S[i][0], \dots, S[i][M - 1]$, **without spaces**).

The output file is made up of a single line containing the return value of function `pulisci`.

Constraints

- $2 \leq N, M \leq 2000$.
- $S[i][j] = '0'$ or $S[i][j] = '1'$ for each $0 \leq i < N, 0 \leq j < M$.

Scoring

Your program will be tested on a number of testcases grouped in subtasks. In order to obtain the score associated to a subtask, you need to correctly solve all the testcases it contains.

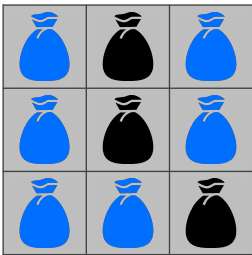
- **Subtask 1** [0 points]: Sample cases.
- **Subtask 2** [7 points]: $N = 2, M = 2$.
- **Subtask 3** [11 points]: $N = 2$.
- **Subtask 4** [16 points]: $N \leq 5, M \leq 5$.
- **Subtask 5** [24 points]: $N \leq 100, M \leq 100$.
- **Subtask 6** [42 points]: No additional constraints.

Examples

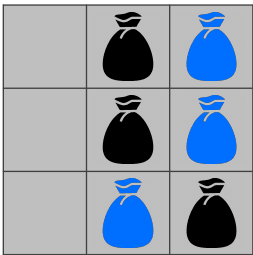
stdin	stdout
3 3 101 101 110	6
3 3 100 100 111	0
4 5 00010 01110 01000 11110	4

Explanation

In the **first sample case**, Edoardo can remove all bags in the left-most column (which contain plastic, in blue), thereafter he won't be able to carry any more bags, and will remain with 6 bags in total.



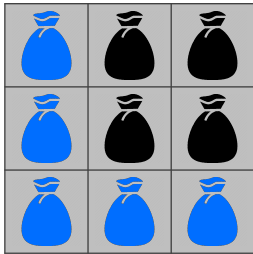
(a) Initial situation



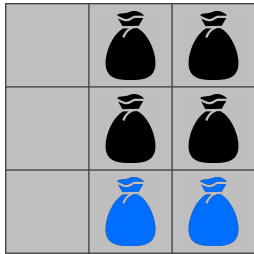
(b) After the first (and only) trip

In the **second sample case** Edoardo can remove all bags by following this strategy:

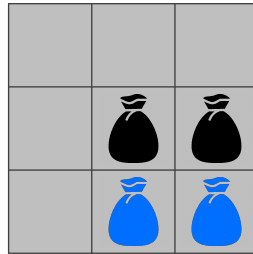
- with the first trip he removes all bags in the column 0 (the left-most), that contain plastic (in blue);
- with the second trip he removes the two remaining bags in row 0, that contain paper (in black);
- with the third trip he removes the two remaining bags in row 1, that contain paper;
- with the fourth and last trip he removes the two remaining bags in row 2, that contain plastic.



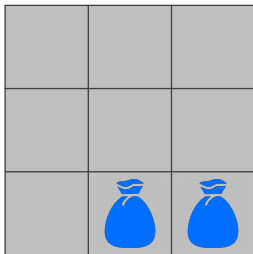
(a) Initial situation



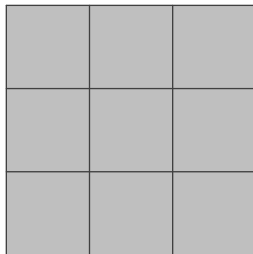
(b) After the first trip



(c) After the second trip



(d) After the third trip



(e) After the fourth (and last) trip