Binary Grid (binarygrid)

You are given a grid A consisting of N rows and M columns, containing only 0s and 1s.

You can do the following operations on A:

- Select a row i $(0 \le i \le N-1)$ and invert it (i.e. flip every value in that row, such that each 0 becomes a 1 and each 1 becomes a 0).
- Select a column j $(0 \le j \le M-1)$ and invert it.

A binary grid is **beautiful** if there are no three consecutive equal values in the same row or in the same column. More formally, there is no i, j $(0 \le i \le N-1, 0 \le j \le M-3)$ such that $A_{i,j} = A_{i,j+1} = A_{i,j+2}$, and there is no i, j $(0 \le i \le N-3, 0 \le j \le M-1)$ such that $A_{i,j} = A_{i+1,j} = A_{i+2,j}$.

Your task is to decide whether it is possible to make a given grid beautiful, and if so, then report the minimum number of operations to do it.

☞ Among the attachments of this task you may find a template file `binarygrid.*` with a sample incomplete implementation.

Input

The first line of the input file contains a single integer T, the number of testcases. T testcases follow, each preceded by an empty line.

Each testcase consists of:

- a line containing two space-separated integers N and M.
- N lines, the $(i+1)$-th of which consisting of string A_i consisting of 0s and 1s representing row i of the grid.

Output

The output file must contain T lines, each consisting of a single integer, the answers of the testcases. If it is possible to make a grid beautiful, then the answer to the testcase is the minimum number of operations to do so, otherwise, if it is impossible, then the answer is `-1`.

Constraints

- $1 \le T \le 100$.
- $1 \le N \le 2000$.
- $1 \le M \le 2000$.
- $A_{i,j}$ is either 0 or 1 for each $i = 0 \ldots N-1$ and $j = 0 \ldots M-1$.
- The sum of N over all testcases is at most 2000.
- The sum of M over all testcases is at most 2000.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 1** (0 points) Examples.
- **Subtask 2** (9 points) $N \leq 10$ and $M \leq 10$. The sum of N over all testcases does not exceed 10 and the sum of M over all testcases does not exceed 10.
- **Subtask 3** (12 points) $N = 1$.
- **Subtask 4** (20 points) $N \leq 10$. The sum of N over all testcases does not exceed 10.
- **Subtask 5** (59 points) No additional constraints.

Examples

input	output

3
 	
4 4
0001
1110
1010
1000
 	
3 3
011
101
110
 	
5 5
11111
10001
11011
10001
11111
	3
	0
	-1

Explanation

Explanation of the first sample:

In the **first testcase**, a possible way to make the grid beautiful using 3 operations is as follows:

- Invert column 0.
- Invert row 2.
- Invert column 1.

In the **second testcase**, the grid is already beautiful thus no operations are needed.

In the **third testcase**, it is impossible to make the grid beautiful using the mentioned operations.