BLOCK_D SOLVER
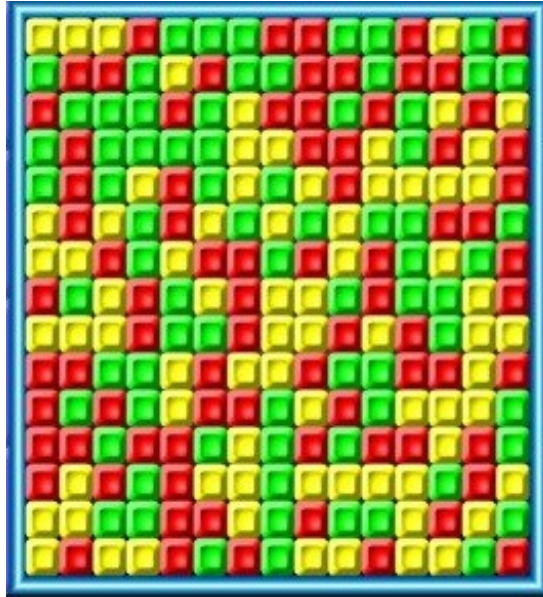
You are given a board of order m×n full of coloured blocks.

At each move, you have to select a block of any colour which has atleast one of it's immediate neighbours of same colour. If you select a block, all the connected blocks(not just the immediate neighbours) of same colour gets destroyed. A block is connected to all the blocks that share an edge with it. Any block will fall directly down if there is no block immediately below it. If a complete column becomes empty, you must either push all the columns to it's left once in the right side or push all the columns to it's right once in the left side(These pushes are not considered as a moves. At each move, you will destroy more than one block).

You will win the game if the remaining number of blocks becomes zero after making the moves. Find the minimum number of moves to win the game. If it is impossible to win the game, print -1.

For better understanding of gameplay you may have a look at this [video](). (optional)

Input:

The first line consists of an integer t, the number of test cases. For each test case the first line consists of two integers m, n the number of rows and columns respectively followed by the matrix representing the coloured blocks. Colour[i][j] contains any character between 'a' and 'e' inclusive.

Output:

For each test case print the minimum number of moves to win the game. If it is impossible to win, print -1.

Input Constraints:

1<=t<=10

1<=m,n<=8

'a'<=Colour[i][j]<='e'

Sample Input:

2

5 5

aaaba

aaaba

aaaba

aaaba

aaaba

5 5

bbbbb

baabb

bbbab

bbbaa

aabab

Sample Output:

2

3