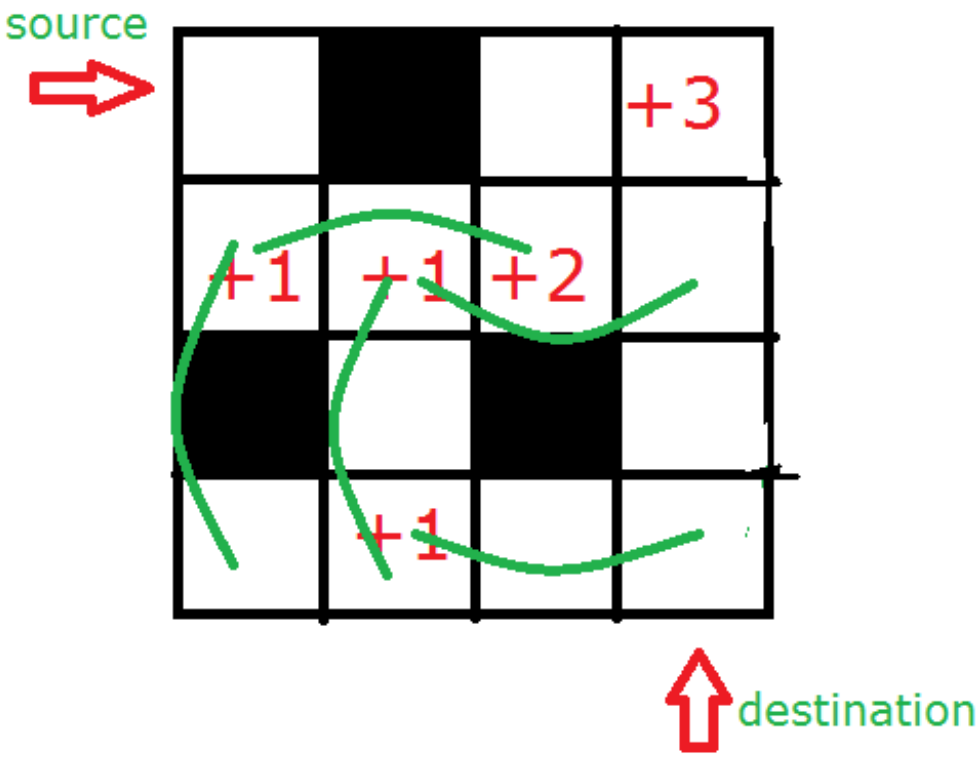
JUMPING DORA

Problem Statement:

Jumping Dora wants to reach her destination from the source as soon as possible.
Dora's initial position is (0,0) and the destination is (m-1,n-1)
If Dora is currently at (i,j), she can either move to (i,j+1) or (i,j+1+C[i][j]) or (i+1,j) or (i+1+C[i][j],j)
Dora cannot move to the blocked positions.

Note that you can jump when there are blocks in between.



Find the shortest time required to reach the destination.

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 and n representing the number of rows and columns. Then follows the description of the matrix C.

Output:

For each test case find the shortest time required to reach the destination. If it is impossible, print -1.

Input Constraints:

1<=t<=100
2<=m,n<=100

C[i][j]={ [0-9] , # }

Note: There may be no path to the destination from the Source. In such cases print -1.

C[0][0] and C[m-1][n-1] are always 0.

Sample Input:

1

4 4

0#03

1120

#0#0

0100

Sample Output:

4