

Natalia Plays a Game

When Natalia is not having fun studying Computing Science in Santiago de los Caballeros, she opens up her bag and pulls out an iPad she won at a Programming Olympiad to play a classical maze game.

The maze is a rectangular figure of M rows and N columns. Open areas are represented by a 'O' while closed areas are represented by a '*'. An area is just a 1×1 block inside the maze. There's an entrance area marked by a '\$' and a destination area marked by a '#'. Please note that both the entrance and destination are also open areas.

Once Natalia is located inside an open area, she can decide to move to either cardinal direction (north, south, east or west). In other words, if Natalia is located at (x,y) , she can move to either $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$ or $(x, y - 1)$. Of course, she can't move inside a closed area.

Given a rectangular maze as described above, output the minimum number of steps necessary to reach the final position from the starting area, if it is possible. If not, output '-1'.

Input

The input contains many lines. The first line contains an integer T ($1 \leq T \leq 100$), the number of test cases. For each test case there's a first line that contains two space separated integers M and N ($1 \leq M \leq 100$), ($2 \leq N \leq 100$), the dimensions of the maze. The line is followed by M lines. Each of these lines contain a string of N characters. The position at index j of the i th string represents the marker of the i,j area. It can be either an open marker ('O'), a closed marker ('*'), the unique entrance marker ('\$') or the unique destination marker ('#').

Output

For each test case output the minimum number of steps necessary to reach the final position from the entrance position. If it is impossible to reach the final position, output -1.

Example

Input:

```
2
3 3
$OO
***
OO#
3 3
$*#
O*O
OOO
```

Output:

```
-1
6
```