Tourist

A lazy tourist wants to visit as many interesting locations in a city as possible without going one step further than necessary. Starting from his hotel, located in the north-west corner of city, he intends to take a walk to the south-east corner of the city and then walk back. When walking to the south-east corner, he will only walk east or south, and when walking back to the north-west corner, he will only walk north or west. After studying the city map he realizes that the task is not so simple because some areas are blocked. Therefore he has kindly asked you to write a program to solve his problem.

Given the city map (a 2D grid) where the interesting locations and blocked areas are marked, determine the maximum number of interesting locations he can visit. Locations visited twice are only counted once.

Input

The first line in the input contains the number of test cases (at most 20). Then follow the cases. Each case starts with a line containing two integers, W and H ($2 \le W$, $H \le 100$), the width and the height of the city map. Then follow H lines, each containing a string with W characters with the following meaning:

. Walkable area
* Interesting location (also walkable area)
# Blocked area

You may assume that the upper-left corner (start and end point) and lower-right corner (turning point) are walkable, and that a walkable path of length H + W – 2 exists between them.

Output

For each test case, output a line containing a single integer: the maximum number of interesting locations the lazy tourist can visit.

Example

Input:
```
2
9 7
*........
.....**#.
..**...#*
..####*#.
.*.#*.*#.
...#**...
*........
5 5
.*.*.
*###.
*.*.*
.###*
.*.*.
```

Output:
7
8