

Labyrinth

(file name: labyrinth.c, labyrinth.p, labyrinth.C)

The northern part of the Pyramid contains a very large and complicated labyrinth. The labyrinth is divided into square blocks, each of them either filled by rock, or free. There is also a little hook on the floor in the center of every free block. The ACM have found that two of the hooks must be connected by a rope that runs through the hooks in every block on the path between the connected ones. When the rope is fastened, a secret door opens. The problem is that we do not know which hooks to connect. That means also that the necessary length of the rope is unknown. Your task is to determine the maximum length of the rope we could need for a given labyrinth.

Input Specification

The input consists of T test cases. The number of them (T) is given on the first line of the input file. Each test case begins with a line containing two integers C and R ($3 \leq C, R \leq 1000$) indicating the number of columns and rows. Then exactly R lines follow, each containing C characters. These characters specify the labyrinth. Each of them is either a hash mark (#) or a period (.). Hash marks represent rocks, periods are free blocks. It is possible to walk between neighbouring blocks only, where neighbouring blocks are blocks sharing a common side. We cannot walk diagonally and we cannot step out of the labyrinth.

The labyrinth is designed in such a way that there is exactly one path between any two free blocks. Consequently, if we find the proper hooks to connect, it is easy to find the right path connecting them.

Output Specification

Your program must print exactly one line of output for each test case. The line must contain the sentence "Maximum rope length is X ." where X is the length of the longest path between any two free blocks, measured in blocks.

Sample Input

```
2
3 3
###
#.#
###
7 6
#####
#.#.###
#.#.###
#.#.#.#
#.....#
#####
```

Output for the Sample Input

```
Maximum rope length is 0.
Maximum rope length is 8.
```