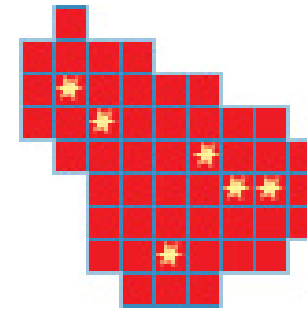
Within a mountain, a diamond deposit is to be recovered which is located in one and the same geological stratum. With regard to this stratum a gridded seismic survey exists. All the grid squares with supposed diamond deposits are marked in this plan.

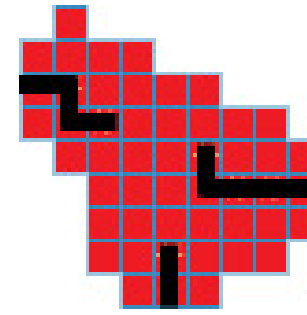
Example:



In order to recover the diamond material, mines have to be dug along the stratum from the outside. Using the machinery available, any mine may only be continued from one grid square to the adjacent square in either a northward or eastward or southward or westward direction.

For safety reasons, mines are not permitted to branch out; so within a mine there may not be any grid square with more than two directly connected grid squares. Note that therefore it is still allowed to have two mines running parallel. Drilling is to proceed economically. A solution is called for that allows access to all diamond deposits while involving the lowest possible number of grid squares — including the places of deposits.

Here is a possible solution for the example:



Input

The first line of input contains a number $T \leq 10$ giving the number of cases to be dealt with. The following lines form the input for the T cases, each in the format described below.

The first line of each test case contains two integers n, m with $3 \leq n, m \leq 11$. The following n lines contain m characters, each of the set {`*`, `.`, `#`}. '`*`' denotes the place of a deposit, '`.`' is another grid square of the plan, and 'tt #' is used as a border and for aligning the grid squares. One can assume that there are between 1 and 10 places of deposits in a given plan, and the plan is entirely surrounded by '`#`'.

Output

For each test case print in a single line the lowest number of grid squares involved in a solution that allows access to all diamond deposits. It can be assumed that there is always a solution, and there are at most 20 grid squares needed.

Sample Input

```
3
11 11
###########
##.########
#....######
#.*....####
#..*.....##
##....*...#
###....**.#
###.......#
###..*...##
####...####
###########
3 11
###########
#..*..*...#
###########
11 11
###########
##.########
#....######
#.*....####
#..*.....##
##....*...#
###....**.#
###....*..#
###..*...##
####...####
###########
```

Sample Output

```
11
2
13
```