

# The Proper Key

Many people think that Tetris was invented by two Russian programmers. But that is not the whole truth. The idea of the game is very old -- even the Egyptians had something similar. But they did not use it as a game. Instead, it was used as a very complicated lock. The lock was made of wood and consisted of a large number of square fields, laid out in regular rows and columns. Each field was either completely filled with wood, or empty. The key for this lock was two-dimensional and it was made by joining square parts of the same size as the fields of the lock. So they had a 2D lock and 2D key that could be inserted into the lock from the top. The key was designed so that it was not possible to move it upwards. It could only fall down and it could slide sideways -- exactly like in a Tetris game. The only difference is that the key could not be rotated. Rotation in Tetris is really a Russian invention.

The entry gate into the Pyramid has such a lock. The ACM archaeologists have found several keys and one of them belongs to the lock with a very high probability. Now they need to try them out and find which one to use. Because it is too time-consuming to try all of them, it is better to begin with those keys that may be inserted deeper into the lock. Your program should be able to determine how deep a given key can be inserted into a given lock.

## Input

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  $R$  and  $C$  ( $1 \leq R, C \leq 100$ ) indicating the key size. Then exactly  $R$  rows follow, each containing  $C$  characters. Each character is either a hash mark (#) or a period (.). A hash mark represents one square field made of wood; a period is an empty field. The wooden fields are always connected, i.e. the whole key is made of one piece. Moreover, the key remains connected even if we cut off arbitrary number of rows from its top. There is always at least one non-empty field in the top-most and bottom-most rows and the left-most and right-most columns.

After the key description, there is a line containing two integers  $D$  and  $W$  ( $1 \leq D \leq 10000$ ,  $1 \leq W \leq 1000$ ). The number  $W$  is the lock width, and  $D$  is its depth. The next  $D$  lines contain  $W$  characters each. The character may be either a hash mark (representing the wood) or a period (the free space).

## Output

Your program should print one line of output for each test case. The line should contain the statement "The key falls to depth  $X$ ". Replace  $X$  with the maximum depth to which the key can be inserted by moving it down and sliding it to the left or right only. The depth is measured as the distance between the bottom side of the key and the top side of the lock. If it is possible to move the key through the whole lock and take it away at the bottom side, output the sentence "The key can fall through."

## Example

Sample Input:

```
4
2 4
```

```
###
###.
3 6
#...#
#...#
#...##
2 3
##.
.#
2 7
#.#.#
.#.#.#
1 1
#
1 10
###...###
3 2
##
.#
.#
1 5
#.#.#
```

Sample output:

The key falls to depth 2.  
The key falls to depth 0.  
The key can fall through.  
The key falls to depth 2.

**Warning: large Input/Output data, be careful with certain languages**