

# Map Exploration Cost

Duck is a young boy who likes to explore the world. Recently, he found an underground place on the Internet and plans to visit. But it is not an easy task because he will have to excavate a tunnel which is a very physical work. Whenever Duck moves one unit, his energy is reduced by one unit, so the exploration cost is increased by one. Luckily, there may be exactly one (or no) existing tunnel or underground space, in this case, Duck can move inside that region without excavating, hence no exploration cost is increased.

You are given an underground map, which is represented as a matrix of **N** rows and **M** columns. The matrix consists of '#' (Rock, visit to there requiring excavation and increase 1 cost), '.' (existing tunnel or space, move between them without increasing cost), 'S' (Starting point), and 'F' (Destination). For example, from # to # / # to . / . to # / S to . / S to # / . to F / # to F / S to F will increase 1 unit of cost, only from . to . will increase no cost.

Given the maximum cost **D**, your task is divided into two parts: 1. Check whether it is possible to reach F from S without spending more than D cost. 2. Output the new map, keep showing 'S' and 'F', and the cells that can be visited within the minimum cost required to reach F (inclusive) are represented by '.', otherwise '#'. Because Duck is not allowed to pass through F, in some cases, it is impossible to get to some cells. Those cells will be represented by '#' as well. Initially, the exploration cost at S is 0, and Duck can only move in four directions (Up, Down, Left, Right), no diagonal move is allowed.

## Input

The first line is the number of test cases **T**. ( $1 \leq T \leq 40$ )

For each test case, it starts with three integers **N** (Number of rows), **M** (Number of columns), **D** (Maximum cost Duck can spend). ( $1 \leq N, M \leq 200, 0 \leq D \leq M + N$ )

The following N lines, each line consisting of M characters, either **S**, **F**, **#** or **.**, representing the map.

It is guaranteed that no two separated tunnels are allowed (at most one, all '.' cells are connected with at least one adjacent edge, or no '.' exists at all), and N and M will not both equal 1.

## Output

For each test case, in the first line, output the word POSSIBLE or IMPOSSIBLE indicating if it is possible to reach F from S within D cost (inclusive).

Then, output N lines containing M characters (S, F, #, .) representing the new map, indicating which cells can be visited without exceeding the minimum cost required to reach F, which not necessarily equal D.

Print a blank line after each test case.

## Example

**Input:**

3

```

8 8 8
#####
#####S##
#.#####
#...#.#
###...##
#####.#
#####
#####F##

```

```

1 10 6
S####.F###

```

```

6 9 4
#####
#S###...#
#####.#
#.#.....
#####
#####F##

```

**Output:**

POSSIBLE

```

#.#.....
....S..
.....
.....
.....
#.#.....
###...#
#####F##

```

POSSIBLE

S.....F###

IMPOSSIBLE

```

.....
.S.....
.....
.....
.....
.....F..

```

**Explanation**

In case 1, the cost matrix is:

- 54432123
- 43321012
- 32332123
- 32223223
- 43322234
- 54433234
- 65544345
- 76655456

We can see that the minimum cost required to reach F from S is 4, which is less than 8, so output POSSIBLE.

For the new map, we keep the S and F, and those cells with minimum cost less than or equal to 4 is represented by '.', otherwise '#'.

In case 2, the cost matrix is  $0123456-1-1-1$ , because we cannot pass through F, but it is impossible to reach the three most right cells without passing through F, so the minimum cost for those cells are both -1.

In case 3, the cost matrix is:

```
212344445
101233334
212344434
323333333
434444444
545555555
```

The answer is impossible because the minimum cost at F is 5 which is more than 4.

However, no cell has minimum cost more than 5, so they are all represented by '.', except S and F.