**Code Jam 2009 - Round 2**

# A Digging Problem

## Problem

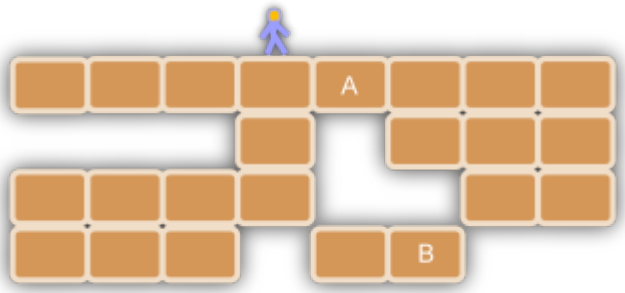
The cave is on fire and there is smoke everywhere! You are trying to dig your way to the bottom of the cave where you can breathe. The problem is that there are some air holes inside the cave, and you don't want to fall too much or you will get hurt.

The cave is represented as an **R** x **C** matrix with air holes and solid rock cells. You start at position (1, 1), which is in the top-left corner. You can move one cell at a time, left or right, if that cell is empty (an air hole). After moving, if the cell below is empty, you fall down until you hit solid rock or the bottom of the cave. The falling distance must be at most **F**,or you will get hurt. You must reach the bottom of the cave without getting hurt. While falling you cannot move left or right.

You can also "dig", turning a cell that contains solid rock into an air hole. The cell that you dig can be one of two cells: the one to your right and below, or the one to your left and below. The cell above the one you are digging has to be empty. While falling you cannot dig.

Your goal is not only to get to the bottom of cave, but also to "dig" as few cells as possible.

Let's describe the operations with a concrete example:




You start at (1, 1) and move right 3 times to position (1, 4), just like the picture.
You dig the rock at position (2, 5). Cell "A" becomes empty.
You move right one position and since there is no cell below you fall 3 cells to position (4, 5).
You dig the rock at position (5, 6). Cell "B" becomes empty.
You move right one position and since there is no cell below you fall 1 cell to position (5, 6).
You have reached the bottom of the cave by digging 2 cells.

## Input

The first line of input gives the number of cases, **N**. **N** test cases follow. The first line of each case is formatted as

```
R C F
```

where **R** is the number of rows in the cave, **C** is the number of columns in the cave, and **F** is the maximum distance you can fall without getting hurt.
This is followed by **R**, rows each of which contains **C** characters. Each character can be one of two things:

- # for a solid rock
- . for an air hole

The top-left cell will always be empty, and the cell below it will be a solid rock.

## Output

For each test case, output one line in the format

```
Case #X: No/Yes [D]
```

where **X** is the case number, starting from 1. Output "No" if you cannot reach the bottom of the cave. Output "Yes **D**" if the bottom of the cave can be reached and the minimum number of cells that need digging is **D**.

## Limits

Memory limit: 1 GB.
$1 \leq \mathbf{N} \leq 50$
$1 \leq \mathbf{F} < \mathbf{R}$

## Small dataset

Time limit: 40 seconds.
$2 \leq \mathbf{R} \leq 10$
$2 \leq \mathbf{C} \leq 6$

## Large dataset

Time limit: 60 seconds.
$2 \leq \mathbf{R} \leq 50$
$2 \leq \mathbf{C} \leq 50$

## Sample

### Sample Input

```
3
2 2 1
.#
##
3 3 1
...
###
###
3 2 1
..
#.
..
```

### Sample Output

```
Case #1: No
Case #2: Yes 3
Case #3: No
```