## Harrichu and the maze P15601_en

Harrichu, the grandson of the famous archaeologist Indiana Jones, is an old man. He feels that nowadays, at his age, it does not have any sense to be the whole day going round mazes up and down, having a leather sofa so comfortable like the one that he has in his office and a middle emptied bottle of chivas.

After consulting it with his friend Jorge Minish, a reputed expert of computing, he has decided that the next maze will be gone around by a scholarship holder,while Harrichu will tell him with SMS which way he must follow. To guide his assistant, our no-so-intrepid archaeologist has a map of the maze.

–In fact –says Jorge while they comment the idea in the luxurious office of Harrichu– you do not even need to guide the scholarship holder. It is very easy to write a program that solves the maze.

–¿Really? –answers Harrichu thoughtful. On the one hand it is too much trouble writing so many SMS, but on the other hand he is worried about how easy is to be replaced by a machine— Mmmm. Actually, I do not need anything as complex as you propose... But, a program that told me if a maze has or not solution would be useful, this way I would only spend my valuable time in the profitable mazes.

¿Can you help Harrichu writing that program?

### Input

The input consists of a line with a number $n$ of mazes, between 1 and 100, followed by the $n$ mazes. A maze is given by two numbers $r$ and $c$ in a line, separated by a space, that indicate the number of rows and columns that have the maze. It is fulfilled that $4 \leq f, c \leq 30$. Afther that, $r$ rows of come with exactly $c$ characters each one, describing the $r$ rows of the maze.

In the description of a maze (look the input instances proposed), a character '.' indicates a square of the maze which the scholarship holder can pass, while a character '#' or an uppercase letter indicates a wall or a trap, squares which the scholarship holder cannot pass. In each maze appears only once the character 'b' (inidicating the initial position of the scholarship holder) and the character 'g' (the position that the scholarship holder must arrive to).

We assure you that the extremes of the maze (that is, the squares that belong to the first or last row or column) are covered by characters '#'.

The scholarship holder can move through the maze with horizontal or vertical steps, but not in diagonal.

Your program must solve an input as the proposed one in 1 second of time.

### Output

The output consists of $n$ lines. For each maze, it must print "Good" if it is solvable (that is, if the scholarship holder can go from 'b' to 'g' doing horizontal or vertical steps and only going through squares '.') and "Bad" if it is not.

## Sample input 1

```
3
6 4
####
#g.#
#H.#
#..#
#b##
####
9 8
########
#.g#..##
#.#....#
#...Y#.#
####...#
##....##
#...b..#
##....##
########
7 7
#######
##S...#
#...#.#
#.bQB.#
####.##
#.g..M#
#######
```

## Sample output 1

```
Good
Good
Bad
```

## Problem information

Author: Omer Giménez
Translator: Carlos Molina

Generation: 2026-01-25T10:12:19.180Z