

# Count Luck



Ron and Hermione are deep in the Forbidden Forest collecting potion ingredients, and they've managed to lose their way. The path out of the forest is blocked, so they must make their way to a portkey that will transport them back to Hogwarts.

Consider the forest as an  $N \times M$  grid. Each cell is either empty (represented by `.`) or blocked by a tree (represented by `X`). Ron and Hermione can move (together inside a single cell) *LEFT*, *RIGHT*, *UP*, and *DOWN* through empty cells, but they *cannot* travel through a tree cell. Their starting cell is marked with the character `M`, and the cell with the portkey is marked with a `*`. The upper-left corner is indexed as  $(0, 0)$ .

```
.X.X.....X
.X*.X.XXX.X
.XX.X.XM...
.....XXXX.
```

In example above, Ron and Hermione are located at index  $(2, 7)$  and the portkey is at  $(1, 2)$ . Each cell is indexed according to [Matrix Conventions](#).

Hermione decides it's time to find the portkey and leave. They start along the path and each time they have to choose a direction, she waves her wand and it points to the correct direction. Ron is betting that she will have to wave her wand exactly  $k$  times. Can you determine if Ron's guesses are correct?

The map from above has been redrawn with the path indicated as a series where `M` is the starting point (no decision in this case), `1` indicates a decision point and `0` is just a step on the path:

```
.X.X.10000X
.X*0X0XXX0X
.XX0X0XM01.
...100XXXX.
```

There are three instances marked with `1` where Hermione must use her wand.

**Note:** It is guaranteed that there is only one path from the starting location to the portkey.

## Function Description

Complete the `countLuck` function in the editor below. It should return a string, either **Impressed** if Ron is correct or **Oops!** if he is not.

`countLuck` has the following parameters:

- *matrix*: a list of strings, each one represents a row of the matrix
- *k*: an integer that represents Ron's guess

## Input Format

The first line contains an integer  $t$ , the number of test cases.

Each test case is described as follows:

The first line contains **2** space-separated integers  $n$  and  $m$ , the number of forest matrix rows and columns.

Each of the next  $n$  lines contains a string of length  $m$  describing a row of the forest matrix.

The last line contains an integer  $k$ , Ron's guess as to how many times Hermione will wave her wand.

## Constraints

- $1 \leq t \leq 10$
- $1 \leq n, m \leq 100$

- $0 \leq k \leq 10000$
- There will be exactly one  $M$  and one  $*$  in the forest.
- Exactly one path exists between  $M$  and  $*$ .

### Output Format

On a new line for each test case, print **Impressed** if Ron impresses Hermione by guessing correctly. Otherwise, print **Oops!**.

### Sample Input

```
3
2 3
*.M
.X.
1
4 11
.X.X.....X
.X*.X.XXX.X
.XX.X.XM...
.....XXXX.
3
4 11
.X.X.....X
.X*.X.XXX.X
.XX.X.XM...
.....XXXX.
4
```

### Sample Output

```
Impressed
Impressed
Oops!
```

### Explanation

For each test case, *count* denotes the number of times Hermione waves her wand.

Case 0: Hermione waves her wand at  $(0, 2)$ , giving us  $count = 1$ . Because  $count = k = 1$ , we print **Impressed** on a new line.

Case 1: Hermione waves her wand at  $(2, 9)$ ,  $(0, 5)$ , and  $(3, 3)$ , giving us  $count = 3$ . Because  $count = k = 3$ , we print **Impressed** on a new line.

Case 2: Hermione waves her wand at  $(2, 9)$ ,  $(0, 5)$ , and  $(3, 3)$ , giving us  $count = 3$ . Because  $count = 3$  and  $k = 4$ ,  $count \neq k$  and we print **Oops!** on a new line.