

# Pointers

You are given an  $R$  rows  $\times$   $C$  columns grid map where each cell contains an arrow (one of the following: '^', '>', '<', or 'v'). If one stands on a cell, then he/she has to go to the neighbouring cell pointed by the arrow in the cell where he/she stands. Supposed you are at cell  $(r, c)$  – row  $r$  and column  $c$ , then:

- '^' means you should go to cell  $(r-1, c)$ .
- '>' means you should go to cell  $(r, c+1)$ .
- '<' means you should go to cell  $(r, c-1)$ .
- 'v' means you should go to cell  $(r+1, c)$ .

If the new cell is out of the grid map, then you fall out of the map.

Your task is to modify the arrows, such that if you start from any cell in the map and follow the arrows, then you will get back to the starting point. Determine the minimum number of arrows that you will have to change.

For example, consider the following map of  $4 \times 2$ .

>v

v<

>v

><

There are several ways to accomplish our goal; here are 3 solution examples:

>< v< >v

>< v^ ^<

>< v^ ><

>< >^ ><

In the first solution example, we have to change 3 arrows:  $\{(1, 2), (2, 1), (3, 2)\}$ . In the second one, we have to change 6 arrows:  $\{(1, 1), (1, 2), (2, 2), (3, 1), (3, 2), (4, 2)\}$ . In the last one, we have to change 2 arrows:  $\{(2, 1), (3, 2)\}$ . There are many other solution examples, but among those, the minimum number of arrows you need to change is 2 (as in solution example 3).

## Input

The first line of input contains an integer  $T$  ( $T \leq 100$ ) denoting the number of cases. Each case begins with two integers  $R$  and  $C$  ( $1 \leq R, C \leq 14$ ) denoting the number of rows and columns of the grid map. The following  $R$  lines each contains  $C$  characters (one of the following: '^', '>', '<', 'v') representing the arrow in each cell.

## Output

For each case, output in a line "Case #X: Y" where X is the case number, starts from 1. and Y is the minimum number of arrows you need to change to accomplish the given goal. Output -1 for Y, if it's not possible to do so.

## Example

### Input:

```
4
4 2
>v
v<
>v
><
3 4
v<>v
v^^v
>^^<
3 4
v<<<
v^v<
>>>^
1 2
>>
```

### Output:

```
Case #1: 2
Case #2: 0
Case #3: 2
Case #4: 1
```

### Explanation:

*Explanation for 2<sup>nd</sup> sample case*

The map already satisfies the goal; there's no need to change any arrow.

*Explanation for 3<sup>rd</sup> sample case*

We need to change at least 2 arrows:

```
v<><
```

```
v^v<
```

```
>^>^
```

*Explanation for 4<sup>th</sup> sample case*

One arrow has to be changed:

```
><
```