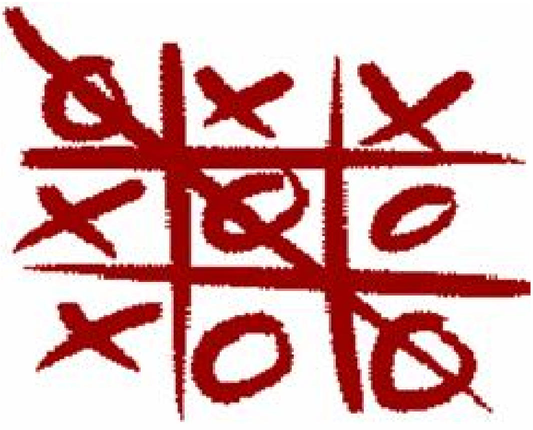
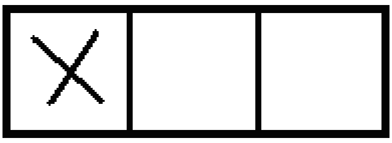


Alice was bored with the game tic-tac-toe. Usually she used to play against computer. She was so bored because, she always managed to make a tie against computer as she already became an expert in this game. She was also bored about the matching things, that means she does not like to see two adjacent X's or two adjacent O's. So Bob created a new computer game for her.

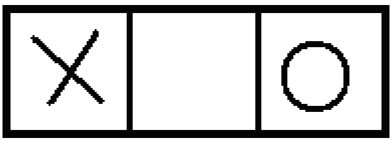
Here is the description of this new 2-player game:

- A 1 × N grid will be given.
- In each turn A player can mark an unmarked cell by an 'X' or by an 'O' (No Zero it is big O).
- No two adjacent X's are allowed in this game.
- No two adjacent O's are allowed in this game.
- The player who marks the last cell wins the game.

Let's have an example for $N = 3$. Suppose first player mark the leftmost cell with an 'X'. Then the grid will look like this -



From this state, the winning move of the second player is to mark the rightmost cell with an 'O'.



Now first player has no move, because in the only empty cell she cannot mark 'X' and cannot mark 'O' because the left of this cell is already marked 'X' and the right of this cell is marked 'O'. The second player will win.

But this is not an optimal example for first player. If in first move, the first player marks the middle cell with any one of 'X' or 'O', she will win.

Now about Bob's computer game, Alice always makes the first move. Alice usually used to start playing with Bob. But most often after some move, Bob gave his responsibility to the computer. Computer always play optimally.

Your are given the state of the grid after Bob's departure and your task is to determine whether it is possible to win against computer from this state for Alice.

Input

The input starts with an integer T ($T < 10001$), which specifies the number of test cases. In each of the next T line there is a string which represents the state of the game after Bob's departure. Here the length of the string is N ($0 < N < 101$). In this string besides 'X' and 'O', another character '.' is used, which represents an unmarked cell.

Output

For each input output a single line, 'Possible.'. If it is possible to win against computer, 'Impossible.' otherwise.

Sample Input

```
3
...
X..
O
```

Sample Output

```
Possible.
Impossible.
Possible.
```