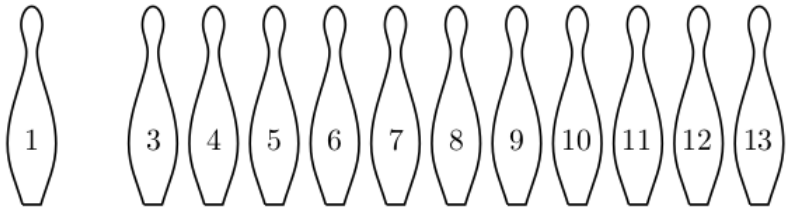
Bowling Pins

Bowling refers to a sport in which a player rolls or throws a bowling ball towards a group of pins. The target is usually to knock down the pins at the end of a lane.

Rules have been slightly modified for this problem. Now, there are n pins, and the pins are arranged in a horizontal line instead of a triangular formation. Two players have to play this game, and they alternate their turns. Whoever knocks down the last pin(s) will be declared the winner.

You and your friends are playing this game. Both of you have become proficient at the sport and can knock down any single pin, or any two adjacent pins at one throw of a bowling ball, however, these are the only moves that you can perform. Some moves have already been played. Suddenly, you realize that it is possible to determine whether this game can be won or not, assuming optimal play. And luckily it's your turn right now.

A configuration is represented by a string consisting of the letters `X` and `I`, where `I` represents a position containing a pin, and `X` represents a position where a pin has been knocked down. An example of such a configuration is shown in the image below. Here, $n = 13$, and the 2^{nd} pin has already been knocked down.



Its representation will be `IXIIIIIIIIIII`.

You are given the current configuration of the pins. If both of you play optimally, determine whether you will win this game or not.

Note

- A player has to knock down at least one pin in his turn.
- Both players play optimally.

Input Format

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

For each test case, the first line contains a single integer n. The second line contains a string of n letters, where each letter is either `I` or `X`.

Constraints

- $1 \leq t \leq 1000$
- $1 \leq n \leq 300$
- Each letter of the string (representing the configuration) is either `I` or `X`.
- There will be at least one `I` in the string.

Output Format

For each test case, print a single line containing `WIN` if you will win this game, otherwise `LOSE`.

Sample Input 0

```
4
4
IXXI
4
XIIX
5
IIXII
5
IIIII
```

Sample Output 0

```
LOSE
WIN
LOSE
WIN
```

Explanation 0

Test Case 1: As the 2 pins are not adjacent, they can't be knocked down in a single turn. Therefore, you can only knock down one of the two pins. Then, in the next turn, your friend will knock down the last pin.
Test Case 2: You can knock down both pins in a single turn.
Test Case 3: You can knock down one or two pins from either side. Your friend can just copy your move on the other side and will be able to get the last move, hence win the game.
Test Case 4: You can knock the middle pin, thus resulting in the configuration `IIXII` for your friend. Now, this configuration is the same as the previous test case. The difference is that now it is your friend's turn and you can copy his shot on the other side.