

Tic-tac-toe is one of the oldest games of mankind. The first records of this game are from the first century BC, during the Roman Empire. John and Mary love to play the game, but after a while they decided to play a variant of the old traditional game, *Tic-tac-toe 1-D*.

Tic-tac-toe 1-D is a game played by two players on a board $1 \times N$; initially, all the squares of the board are empty. Players take turns drawing a cross on an empty square. The first player to complete a sequence of three or more crosses in contiguous squares wins the game.

Mary soon realized that, depending on the game situation, being her turn, she can guarantee she will win, regardless of John's moves. This is relatively easy for smaller boards, but for larger boards, after several moves, this task is more difficult. So, she asked you to write a program that, given the state the board, decides whether there exists a winning strategy.

Input

The input contains several test cases. The first line of a test case contains an integer N , indicating the size of the board ($3 \leq N \leq 10^4$). The next line contains a sequence of N characters indicating which squares of the board have been marked: a '.' indicates that the corresponding square is empty, while a 'X' indicates that the square already has a cross drawn. The input never contains three contiguous 'X'.

The last test case is followed by a line containing a single number zero.

Output

For each test case in the input your program must print a single line containing a single character, 'S' if Mary has a winning strategy or 'N' otherwise.

Sample Input

```
5
.....
5
..X..
6
X.X.X.
12
.....
0
```

Sample Output

```
S
N
S
N
```