Problem C
Stephen Cook

Time limit: 3 seconds
Memory limit: 256 megabytes

Problem Description

> Stephen Arthur Cook is a computer scientist and a mathematician who specializes in hard stuffs such as complexity theory. He gave a precise statement of the P versus NP problem in his paper "The complexity of theorem-proving procedures", and one of his most famous contribution is proving that Boolean satisfiability problem, a.k.a. SAT, is NP-hard.

Cook won the Turing Award in 1982. Yes, that Turing. If you don't know what I am talking about, go back to problem A. Because you are a nerd also named Cook, so you challenge your equally nerdy friend Levin to play a nerdy game inspired by Stephen Cook together. Here is how it works, you are going to prepare a boolean formula (actually we will prepare it for you). And then, you and Levin will take turns in deciding the value of one variable (true or false). Once the value of every variable is decided, if the formula is satisfied, you win! Otherwise, Levin will win. So, are you nerdy enough to tell me if you can win this game if you pick first? Assuming that Levin plays optimally of course.

Input Format

The first line of the input will be a single integer T ($T \leq 20$) indicating the number of test cases. Each test case will consist of two lines.

The first line contains an integer n ($1 \leq n \leq 10$) indicating the number of variables in the boolean formula. The i-th variable will be represented by the i-th English alphabet in upper cases. That is, the first variable is `A`, and the third variable is `C`.

For each test case, one boolean formula will be given in its second line which has length no more than 256 characters. A boolean formula must be in one of the following five forms.

- `var`: `var` is a variable.
- `( formula1 )`: `formula1` is a boolean formula.
- `not formula1`: `formula1` is a boolean formula.
- `formula1 or formula2`: both of `formula1` and `formula2` are boolean formulas.
- `formula1 and formula2`: both of `formula1` and `formula2` are boolean formulas.

You may assume that there are blanks to separate the variables and operators in the input file.

There are four kinds of boolean operators: `and`, `or`, `not`, and brackets `()`. Note that the first three kinds of operator are in lower cases. The operator precedence given from high to low as follows: `()` > `not` > `and` > `or`.

Output Format

Output the winner's name ("`Cook`" or "`Levin`").

Sample Input

```
3
1
A and not A
1
A or not A
3
( A or C ) and B
```

Sample Output

```
Levin
Cook
Cook
```