



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
```

Problem C

Stephen Cook

Time limit: 3 seconds

Memory limit: 256 megabytes

Problem Description

史提芬·亞瑟·庫克 (Stephen Arthur Cook) 是個計算機科學家，專長領域是艱澀的計算複雜度理論。他在論文 “The complexity of theorem-proving procedures” 中提出了精確的 P 對 NP 問題的經確定義，而他最重要的貢獻之一，是證明了 Boolean satisfiability (SAT) 問題為 NP -hard。

庫克為 1982 年圖靈獎得主。是的，就是 Problem A 的那位圖靈。恰好你也是個叫做庫克的宅宅，你的宅宅好朋友列文 (Levin) 跟你挑戰一起玩個由庫克定理啟發宅宅的遊戲。這遊戲是這樣進行的，首先你們準備了一個 Boolean formula (實際上是出題者準備的)，然後你跟列文輪流，每一個回合，挑選一個變數 (variable) 並決定其真假值。直到所有的變數值被設定時，遊戲就結束。如果最後 Boolean formula 運算出來的值為真，則你獲勝，反之列文獲勝。假定你先手，且列文總是用最佳策略進行遊戲。你有宅到可以告訴我，究竟誰會勝利嗎？

Input Format

輸入資料的第一行有一個整數 T ($T \leq 20$) 代表有多少組測試資料。

每一組測試資料有兩行，第一行有一個整數 n ($1 \leq n \leq 10$) 代表 Boolean formula 有多少個變數，其中第 i 個變數會用第 i 個英文字母的大寫代表。即第一個變數為 A、第三個變數為 C。

第二行有一個至多有 256 個字元的 Boolean formula。一個 Boolean formula 必須為下列五種形式之一：

- `var`: `var` 是一個變數。
- `(formula1)`: `formula1` 是一個 Boolean formula。
- `not formula1`: `formula1` 是一個 Boolean formula。
- `formula1 or formula2`: `formula1` 和 `formula2` 都是 Boolean formula。
- `formula1 and formula2`: `formula1` 和 `formula2` 都是 Boolean formula。

你可以假定變數跟運算子之間都有空白隔開。

本題只有似種運算子：`and`、`or`、`not`、括號`()`。前三種都用小寫字母。運算優先順序為：`()` > `not` > `and` > `or`。

Output Format

請輸出一行是誰贏了。請輸出 “Cook” 或 “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
```