A. Little Girl and Game

time limit per test: 2 seconds
memory limit per test: 256 megabytes

The Little Girl loves problems on games very much. Here's one of them.

Two players have got a string s, consisting of lowercase English letters. They play a game that is described by the following rules:

- The players move in turns; In one move the player can remove an arbitrary letter from string s.
- If the player before his turn can reorder the letters in string s so as to get a palindrome, this player wins. A palindrome is a string that reads the same both ways (from left to right, and vice versa). For example, string "abba" is a palindrome and string "abc" isn't.

Determine which player will win, provided that both sides play optimally well — the one who moves first or the one who moves second.

Input

The input contains a single line, containing string s ($1 \le |s| \le 10^3$). String s consists of lowercase English letters.

Output

In a single line print word "First" if the first player wins (provided that both players play optimally well). Otherwise, print word "Second". Print the words without the quotes.

Examples

input	output
aba	First

input	output
abca	Second

B. Yet Another Number Game

time limit per test: 2 seconds
memory limit per test: 256 megabytes

Since most contestants do not read this part, I have to repeat that Bitlandians are quite weird. They have their own jobs, their own working method, their own lives, their own sausages and their own games!

Since you are so curious about Bitland, I'll give you the chance of peeking at one of these games.

BitLGM and BitAryo are playing yet another of their crazy-looking genius-needed Bitlandish games. They've got a sequence of n non-negative integers $a_1, a_2, ..., a_n$. The players make moves in turns. BitLGM moves first. Each player can and must do one of the two following actions in his turn:

- Take one of the integers (we'll denote it as a_i). Choose integer x $(1 \le x \le a_i)$. And then decrease a_i by x, that is, apply assignment: $a_i = a_i - x$.
- Choose integer x $(1 \le x \le \min_{i=1}^{n} a_i)$. And then decrease all a_i by x, that is, apply assignment: $a_i = a_i - x$, for all i.

The player who cannot make a move loses.

You're given the initial sequence $a_1, a_2, ..., a_n$. Determine who wins, if both players plays optimally well and if BitLGM and BitAryo start playing the described game in this sequence.

Input

The first line contains an integer n $(1 \le n \le 3)$.

The next line contains n integers $a_1, a_2, ..., a_n$ $(0 \le a_i < 300)$.

Output

Write the name of the winner (provided that both players play optimally well). Either "`BitLGM`" or "`BitAryo`" (without the quotes).

Examples

input
```
2
1 1
```
output
```
BitLGM
```

input
```
2
1 2
```
output
```
BitAryo
```

input
```
3
1 2 1
```
output
```
BitLGM
```

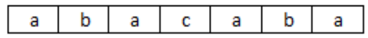
C. Playing with String

time limit per test: 2 seconds

memory limit per test: 256 megabytes

Two people play the following string game. Initially the players have got some string s. The players move in turns, the player who cannot make a move loses.

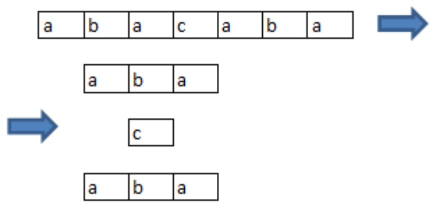
Before the game began, the string is written on a piece of paper, one letter per cell.



An example of the initial situation at s = "abacaba"

A player's move is the sequence of actions:

1. The player chooses one of the available pieces of paper with some string written on it. Let's denote it is t. Note that initially, only one piece of paper is available.
2. The player chooses in the string $t = t_1t_2... t_{|t|}$ character in position i $(1 \le i \le |t|)$ such that for some positive integer l $(0 < i - l;\ i + l \le |t|)$ the following equations hold: $t_{i-1} = t_{i+1}$, $t_{i-2} = t_{i+2}$, ..., $t_{i-l} = t_{i+l}$.
3. Player cuts the cell with the chosen character. As a result of the operation, he gets three new pieces of paper, the first one will contain string $t_1t_2... t_{i-1}$, the second one will contain a string consisting of a single character t_i, the third one contains string $t_{i+1}t_{i+2}... t_{|t|}$.



An example of making action $(i = 4)$ with string s = «abacaba»

Your task is to determine the winner provided that both players play optimally well. If the first player wins, find the position of character that is optimal to cut in his first move. If there are multiple positions, print the minimal possible one.

Input

The first line contains string s $(1 \le |s| \le 5000)$. It is guaranteed that string s only contains lowercase English letters.

Output

If the second player wins, print in the single line "Second" (without the quotes). Otherwise, print in the first line "First" (without the quotes), and in the second line print the minimal possible winning move — integer i $(1 \le i \le |s|)$.

Examples

input	output
abacaba	First 2

input	output
abcde	Second

Note

In the first sample the first player has multiple winning moves. But the minimum one is to cut the character in position 2.

In the second sample the first player has no available moves.

D. 1-2-K Game

time limit per test: 2 seconds
memory limit per test: 256 megabytes

Alice and Bob play a game. There is a paper strip which is divided into $n + 1$ cells numbered from left to right starting from 0. There is a chip placed in the n-th cell (the last one).

Players take turns, Alice is first. Each player during his or her turn has to move the chip 1, 2 or k cells to the left (so, if the chip is currently in the cell i, the player can move it into cell $i - 1$, $i - 2$ or $i - k$). The chip should not leave the borders of the paper strip: it is impossible, for example, to move it k cells to the left if the current cell has number $i < k$. The player who can't make a move loses the game.

Who wins if both participants play optimally?

Alice and Bob would like to play several games, so you should determine the winner in each game.

Input

The first line contains the single integer T ($1 \le T \le 100$) — the number of games. Next T lines contain one game per line. All games are independent.

Each of the next T lines contains two integers n and k ($0 \le n \le 10^9$, $3 \le k \le 10^9$) — the length of the strip and the constant denoting the third move, respectively.

Output

For each game, print `Alice` if Alice wins this game and `Bob` otherwise.

Example

input	output
4 0 3 3 3 3 4 4 4	Bob Alice Bob Alice

E. Tokitsukaze, CSL and Stone Game

time limit per test: 1 second
memory limit per test: 256 megabytes

Tokitsukaze and CSL are playing a little game of stones.

In the beginning, there are n piles of stones, the i-th pile of which has a_i stones. The two players take turns making moves. Tokitsukaze moves first. On each turn the player chooses a nonempty pile and removes exactly one stone from the pile. A player loses if all of the piles are empty before his turn, or if after removing the stone, two piles (possibly empty) contain the same number of stones. Supposing that both players play optimally, who will win the game?

Consider an example: $n = 3$ and sizes of piles are $a_1 = 2$, $a_2 = 3$, $a_3 = 0$. It is impossible to choose the empty pile, so Tokitsukaze has two choices: the first and the second piles. If she chooses the first pile then the state will be $[1, 3, 0]$ and it is a good move. But if she chooses the second pile then the state will be $[2, 2, 0]$ and she immediately loses. So the only good move for her is to choose the first pile.

Supposing that both players always take their best moves and never make mistakes, who will win the game?

Note that even if there are two piles with the same number of stones at the beginning, Tokitsukaze may still be able to make a valid first move. It is only necessary that there are no two piles with the same number of stones after she moves.

Input

The first line contains a single integer n ($1 \le n \le 10^5$) — the number of piles.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_1, a_2, \ldots, a_n \le 10^9$), which mean the i-th pile has a_i stones.

Output

Print "`sjfnb`" (without quotes) if Tokitsukaze will win, or "`cslnb`" (without quotes) if CSL will win. Note the output characters are case-sensitive.

Examples

input
```
1
0
```
output
```
cslnb
```

input
```
2
1 0
```
output
```
cslnb
```

input
```
2
2 2
```
output
```
sjfnb
```

input
```
3
2 3 1
```
output
```
sjfnb
```

Note

In the first example, Tokitsukaze cannot take any stone, so CSL will win.

In the second example, Tokitsukaze can only take a stone from the first pile, and then, even though they have no stone, these two piles will have the same number of stones, which implies CSL will win.

In the third example, Tokitsukaze will win. Here is one of the optimal ways:

- Firstly, Tokitsukaze can choose the first pile and take a stone from that pile.
- Then, CSL can only choose the first pile, because if he chooses the second pile, he will lose immediately.
- Finally, Tokitsukaze can choose the second pile, and then CSL will have no choice but to lose.

In the fourth example, they only have one good choice at any time, so Tokitsukaze can make the game lasting as long as possible and finally win.

F. Game With String

time limit per test: 3 seconds
memory limit per test: 256 megabytes

Alice and Bob play a game. Initially they have a string $s_1, s_2, \dots, s_n$, consisting of only characters . and X. They take alternating turns, and Alice is moving first. During each turn, the player has to select a contiguous substring consisting only of characters . and replaces each of them with X. Alice must select a substing of length a, and Bob must select a substring of length b. It is guaranteed that $a > b$.

For example, if $s =$ `...X..` and $a = 3$, $b = 2$, then after Alice's move string can turn only into `XXXX...` And if it's Bob's turn and the string $s =$ `...X..`, then after Bob's move the string can turn into `XX.X..`, `.XXX..` or `...XXX`.

Whoever is unable to make a move, loses. You have to determine who wins if they both play optimally.

You have to answer q independent queries.

Input

The first line contains one integer q ($1 \le q \le 3 \cdot 10^5$) — the number of queries.

The first line of each query contains two integers a and b ($1 \le b < a \le 3 \cdot 10^5$).

The second line of each query contains the string s ($1 \le |s| \le 3 \cdot 10^5$), consisting of only characters . and X.

It is guaranteed that sum of all $|s|$ over all queries not exceed $3 \cdot 10^5$.

Output

For each test case print YES if Alice can win and NO otherwise.

You may print every letter in any case you want (so, for example, the strings yEs, yes, Yes and YES will all be recognized as positive answer).

Example

input	output
3 3 2 XX......XX...X 4 2 X...X.X..X 5 3 ....X..X	YES NO YES

Note

In the first query Alice can select substring $s_3 \dots s_5$. After that s turns into `XXXXX...XX...X`. After that, no matter what move Bob makes, Alice can make the move (this will be her second move), but Bob can't make his second move.

In the second query Alice can not win because she cannot even make one move.

In the third query Alice can choose substring $s_2 \dots s_6$. After that s turns into `.XXXXX.X..X`, and Bob can't make a move after that.