15-295 Fall 2018 #11 Games and Nimbers

A. Game

0.5 seconds, 64 megabytes

There is a legend in the IT City college. A student that failed to answer all questions on the game theory exam is given one more chance by his professor. The student has to play a game with the professor.

The game is played on a square field consisting of $n \times n$ cells. Initially all cells are empty. On each turn a player chooses and paint an empty cell that has no common sides with previously painted cells. Adjacent corner of painted cells is allowed. On the next turn another player does the same, then the first one and so on. The player with no cells to paint on his turn loses.

The professor have chosen the field size n and allowed the student to choose to be the first or the second player in the game. What should the student choose to win the game? Both players play optimally.

Input

The only line of the input contains one integer n $(1 \leq n \leq 10^{18})$ — the size of the field.

Output

Output number `1`, if the player making the first turn wins when both players play optimally, otherwise print number `2`.

input
1
output
1

input
2
output
2

B. Industrial Nim

2 seconds, 64 megabytes

There are n stone quarries in Petrograd.

Each quarry owns m_i dumpers $(1 \le i \le n)$. It is known that the first dumper of the i-th quarry has x_i stones in it, the second dumper has $x_i + 1$ stones in it, the third has $x_i + 2$, and the m_i-th dumper (the last for the i-th quarry) has $x_i + m_i - 1$ stones in it.

Two oligarchs play a well-known game Nim. Players take turns removing stones from dumpers. On each turn, a player can select any dumper and remove any non-zero amount of stones from it. The player who cannot take a stone loses.

Your task is to find out which oligarch will win, provided that both of them play optimally. The oligarchs asked you not to reveal their names. So, let's call the one who takes the first stone «`tolik`» and the other one «`bolik`».

Input

The first line of the input contains one integer number n $(1 \le n \le 10^5)$ — the amount of quarries. Then there follow n lines, each of them contains two space-separated integers x_i and m_i $(1 \le x_i, m_i \le 10^{16})$ — the amount of stones in the first dumper of the i-th quarry and the number of dumpers at the i-th quarry.

Output

Output «`tolik`» if the oligarch who takes a stone first wins, and «`bolik`» otherwise.

input

```
2
2 1
3 2
```

output

```
tolik
```

input

```
4
1 1
1 1
1 1
1 1
```

output

```
bolik
```

C. Game of Stones

3 seconds, 256 megabytes

Sam has been teaching Jon the *Game of Stones* to sharpen his mind and help him devise a strategy to fight the white walkers. The rules of this game are quite simple:

- The game starts with n piles of stones indexed from 1 to n. The i-th pile contains s_i stones.
- The players make their moves alternatively. A move is considered as removal of some number of stones from a pile. Removal of 0 stones does not count as a move.
- The player who is unable to make a move loses.

Now Jon believes that he is ready for battle, but Sam does not think so. To prove his argument, Sam suggested that they play a modified version of the game.

In this modified version, no move can be made more than once on a pile. For example, if 4 stones are removed from a pile, 4 stones cannot be removed from that pile again.

Sam sets up the game and makes the first move. Jon believes that Sam is just trying to prevent him from going to battle. Jon wants to know if he can win if both play optimally.

Input

First line consists of a single integer n $(1 \le n \le 10^6)$ — the number of piles.

Each of next n lines contains an integer s_i $(1 \le s_i \le 60)$ — the number of stones in i-th pile.

Output

Print a single line containing "YES" (without quotes) if Jon wins, otherwise print "NO" (without quotes)

input

```
1
5
```

output

```
NO
```

input

```
2
1
2
```

output

```
YES
```

In the first case, Sam removes all the stones and Jon loses.

In second case, the following moves are possible by Sam:
$\{1,2\} \to \{0,2\}, \{1,2\} \to \{1,0\}, \{1,2\} \to \{1,1\}$

In each of these cases, last move can be made by Jon to win the game as follows:
$\{0,2\} \to \{0,0\}, \{1,0\} \to \{0,0\}, \{1,1\} \to \{0,1\}$

D. Sagheer and Apple Tree

2 seconds, 256 megabytes

Sagheer is playing a game with his best friend Soliman. He brought a tree with n nodes numbered from 1 to n and rooted at node 1. The i-th node has a_i apples. This tree has a special property: the lengths of all paths from the root to any leaf have the same parity (i.e. all paths have even length or all paths have odd length).

Sagheer and Soliman will take turns to play. Soliman will make the first move. The player who can't make a move loses.

In each move, the current player will pick a single node, take a non-empty subset of apples from it and do one of the following two things:

1. eat the apples, if the node is a leaf.
2. move the apples to one of the children, if the node is non-leaf.

Before Soliman comes to start playing, Sagheer will make **exactly one change** to the tree. He will pick two different nodes u and v and swap the apples of u with the apples of v.

Can you help Sagheer count the number of ways to make the swap (i.e. to choose u and v) after which he will win the game if both players play optimally? (u, v) and (v, u) are considered to be the same pair.

Input

The first line will contain one integer n $(2 \le n \le 10^5)$ — the number of nodes in the apple tree.

The second line will contain n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^7)$ — the number of apples on each node of the tree.

The third line will contain n - 1 integers $p_2, p_3, ..., p_n$ $(1 \le p_i \le n)$ — the parent of each node of the tree. Node i has parent p_i (for $2 \le i \le n$). Node 1 is the root of the tree.

It is guaranteed that the input describes a valid tree, and the lengths of all paths from the root to any leaf will have the same parity.

Output

On a single line, print the number of different pairs of nodes (u, v), $u \ne v$ such that if they start playing after swapping the apples of both nodes, Sagheer will win the game. (u, v) and (v, u) are considered to be the same pair.

input
3 2 2 3 1 1

output
1

input
3 1 2 3 1 1

output
0

input
8 7 2 2 5 4 3 1 1 1 1 1 4 4 5 6

output
4

In the first sample, Sagheer can only win if he swapped node 1 with node 3. In this case, both leaves will have 2 apples. If Soliman makes a move in a leaf node, Sagheer can make the same move in the other leaf. If Soliman moved some apples from a root to a leaf, Sagheer will eat those moved apples. Eventually, Soliman will not find a move.

In the second sample, There is no swap that will make Sagheer win the game.

Note that Sagheer must make the swap even if he can win with the initial tree.

E. Permutation Game

1 second, 256 megabytes

After a long day, Alice and Bob decided to play a little game. The game board consists of n cells in a straight line, numbered from 1 to n, where each cell contains a number a_i between 1 and n. Furthermore, no two cells contain the same number.

A token is placed in one of the cells. They take alternating turns of moving the token around the board, with Alice moving first. The current player can move from cell i to cell j only if the following two conditions are satisfied:

- the number in the new cell j must be strictly larger than the number in the old cell i (i.e. $a_j > a_i$), and
- the distance that the token travels during this turn must be a multiple of the number in the old cell (i.e. $|i - j| \bmod a_i = 0$).

Whoever is unable to make a move, loses. For each possible starting position, determine who wins if they both play optimally. It can be shown that the game is always finite, i.e. there always is a winning strategy for one of the players.

Input

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

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$). Furthermore, there are no pair of indices $i \neq j$ such that $a_i = a_j$.

Output

Print s — a string of n characters, where the i-th character represents the outcome of the game if the token is initially placed in the cell i. If Alice wins, then s_i has to be equal to "A"; otherwise, s_i has to be equal to "B".

input

```
8
3 6 5 4 2 7 1 8
```

output

```
BAAAABAB
```

input

```
15
3 11 2 5 10 9 7 13 15 8 4 12 6 1 14
```

output

```
ABAAAABBBAABAAB
```

In the first sample, if Bob puts the token on the number (**not position**):

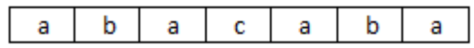
- 1: Alice can move to any number. She can win by picking 7, from which Bob has no move.
- 2: Alice can move to 3 and 5. Upon moving to 5, Bob can win by moving to 8. If she chooses 3 instead, she wins, as Bob has only a move to 4, from which Alice can move to 8.
- 3: Alice can only move to 4, after which Bob wins by moving to 8.
- 4, 5, or 6: Alice wins by moving to 8.
- 7, 8: Alice has no move, and hence she loses immediately.

F. Playing with String

2 seconds, 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.

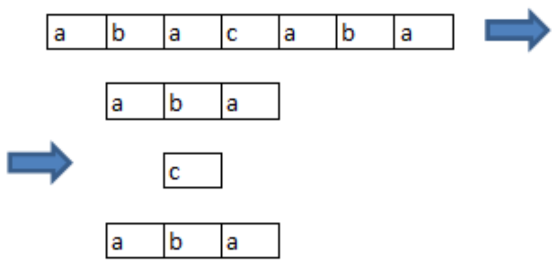
Before the game began, the string is written on a piece of paper, one letter per cell.



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 $t = t_1 t_2 ... t_{|t|}$ 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 $t_1 t_2 ... t_{i-1}$, 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 $(1 \le i \le |s|)$.

input
```
abacaba
```
output
```
First
2
```

input
```
abcde
```
output
```
Second
```

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.