This is an unofficial file made by me with the purpose of compiling all problems into a single file.

The Secret Key of the Ancient Ruins

Description:

As an explorer, you find yourself in ancient ruins, where you encounter a beautifully carved stone pillar inscribed with ancient text, named "T". Legend has it that "T" conceals the key to a mysterious treasure.

Your task is to decipher the secret and find this mystical key. The key must satisfy the following conditions:

1. The key is a prefix of "T".
2. The key is also a suffix of "T".
3. In addition to conditions 1 and 2, the key must appear at least once more somewhere in the middle of "T", i.e., the key is also present in "T", but not as either a prefix of "T" or a suffix of "T" (Note that the key is the longest one).

Input Format:

A single string "T", consisting of lowercase letters only, with a length in the range [1, 1000000].

Output Format:

Print the key "T". If no such key exists, print "Just a legend" (without quotes).

Input Example1:

```
fixprefixsuffix
```

Output Example1:

```
fix
```

Input Example2:

```
abcdabc
```

Output Example2:

```
Just a legend
```

The Maximal Magical Power

Description:

You are a digital magician, set to perform a special number transformation magic using a sequence of n mysterious numbers from an ancient scroll.

The performance unfolds as follows:

1. You read a number n from the ancient scroll, indicating the count of numbers in the sequence.
2. You reveal n mysterious numbers, each imbued with magical power.

3. The grand reveal: You will find two numbers through a mystical numerical transformation ritual, such that their magical power (i.e., the XOR value) is maximized.

Your goal is to demonstrate to the audience the result of maximizing the magical power of these two numbers.

Input Format:

The first line contains an integer n, where $1 \leq n \leq 100,000$. The second line contains n integers, each within the range of $1 \leq$ integer $\leq 100,000$

Output Format:

Print the result after maximizing the magical power.

Input Example 1:

```
4
1 2 3 4
```

Output Example 1:

```
7
```

Input Example 2:

```
4
1 16 17 3
```

Output Example 2:

```
19
```

Input Example 3:

```
7
45 23 58 95 32 17 64
```

Output Example 3:

```
127
```

Input Example 4:

```
4
31 14 15 92
```

Output Example 4:

```
83
```

Military Squad Management

Description:

Assume you are a military commander in charge of a special squad composed of n soldiers, each with a unique identifier ranging from 1 to n. The soldiers are arranged in n single-file columns, with each soldier i standing in the i-th column.

Your task is to perform the following operations on these squads:

- Merge Command "M i j": This command means the entire column in which soldier i is standing (head to tail) is attached to the end of the column in which soldier j is standing. It is guaranteed that soldier i and soldier j are not in the same column at the time of the command.
- Query Command "C i j": This command asks whether soldiers i and j are in the same column. If they are, output the number of soldiers between i and j.

Input Format:

The first line contains an integer T, representing the total number of commands to be executed. The following T lines contain one command each. There are two types of commands:

- "M i j": i and j are integers, representing the merging of the columns of soldiers i and j.
- "C i j": i and j are integers, asking whether soldiers i and j are in the same column.

For 100% of the data, it is guaranteed that:

$$1 \leq T \leq 500,000 \quad \text{and} \quad 1 \leq i, j \leq 30,000.$$

Output Format:

- For merge commands, no output is required.
- For query commands, if soldiers i and j are in the same column, output the number of soldiers between i and j; otherwise, output -1.

Input Example 1:

```
4
M 2 3
C 1 2
M 2 4
C 4 2
```

Output Example 1:

```
-1
1
```

The following table represents the state of the columns after each command:

```
        | Column 1 | Column 2 | Column 3 | Column 4 |
-------------------------------------------------
Initial |    1     |    2     |    3     |    4     |
-------------------------------------------------
M 2 3   |    1     |          |    3     |    4     |
        |          |          |    2     |          |
```

```
-----------------------------------------------------
C 1 2   | Since soldiers 1 and 2 are not in the     |
        | same column, output -1                    |
-----------------------------------------------------
M 2 4   |    1     |          |          |    4     |
        |          |          |          |    3     |
        |          |          |          |    2     |
-----------------------------------------------------
C 4 2   | There is only one soldier (3) between     |
        | soldiers 4 and 2, so output 1             |
-----------------------------------------------------
```

Input Example 2:

```
5
M 1 2
M 3 4
C 1 3
M 2 3
C 1 4
```

Output Example 2:

```
-1
2
```

Input Example 3:

```
4
M 1 3
M 2 4
C 1 4
C 2 3
```

Output Example 3:

```
-1
-1
```

Art Gallery Rearrangement

Description:

You are assisting an interior designer in rearranging the artworks in a square art gallery. The gallery's walls are divided into an $n \times n$ grid, with each cell containing a painting that has a dominant color: either black or white.

Operations are as follows:

- Row Rotation: You can select any two rows in the gallery and swap the positions of all paintings in these rows.
- Column Rotation: You can select any two columns in the gallery and swap the positions of all paintings in these columns.

The designer's goal is to perform several operations so that all paintings along the line of sight from the entrance to the farthest end of the gallery (i.e., the diagonal line from the top-left corner to the bottom-right corner) are of black color.

Input Format:

The first line contains an integer T, representing the number of test cases. For each test case:

- The first line contains an integer n, representing the size of the square matrix.
- The next n lines contain n integers each, representing the matrix. 0 represents white, and 1 represents black.

For 20% of the data, it is guaranteed that $n \leq 7$. For 50% of the data, it is guaranteed that $n \leq 50$. For 100% of the data, it is guaranteed that $1 \leq n \leq 200$ and $1 \leq T \leq 20$.

Output Format:

For each test case, output a single line with the string "Yes" if the arrangement is possible, or "No" if it is not.

Input Example 1:

```
2
2
0 0
0 1
3
0 0 1
0 1 0
1 0 0
```

Output Example 1:

```
No
Yes
```

Input Example 2:

```
2
2
1 0
0 1
3
1 0 0
0 1 0
0 0 1
```

Output Example 2:

```
Yes
Yes
```

Input Example 3:

```
2
4
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
3
0 1 0
1 0 1
0 1 0
```

Output Example 3:

```
Yes
No
```