rot3ad

Yandex Cup 2021 mashup

11 Jan 2023, 17:00:49
start: 9 Nov 2021, 22:00:00
start: 9 Nov 2021, 22:00:00

A. ZeroOne

Time limit	1 second
Memory limit	1024.0 MB
Input	stdin or input.txt
Output	stdout or output.txt

Compare two numbers in binary notation. They're represented by a sequence of words concatenated without spaces (0 — `zero`, 1 — `one`).

Input format

The first line contains the string representation of the number s_1 ($3 \le |s_1| \le 1000$).

The second line contains the string representation of the number s_2 ($3 \le |s_2| \le 1000$).

The numbers do not contain leading zeros.

Output format

Output the character `>` (`ASCII 62`) if the first number is greater than the second, the character `<` (`ASCII 60`) if the second number is greater than the first, otherwise the character `=` (`ASCII 61`).

Sample 1

Input	Output
oneone onezerozero	<

Sample 2

Input	Output
zero zero	=

Sample 3

Input	Output
onezero oneone	<

Sample 4

Input	Output
oneonezerozero onezeroonezero	>

Sample 5

Input	Output
one zero	>

Sample 6

Input	Output
one one	=

Language OpenJDK Java 11

Type here Send file

```java
import java.util.Scanner;

public class Yandex {

    public static void main(String[] args) {
        solveTask();
    }

    static void solveTask() {
        Scanner scanner = new Scanner(System.in);
        String firstStr = scanner.nextLine();
        String secondStr = scanner.nextLine();
        StringBuilder firstNumber = convertFromSymbolsToNumericChars(firstStr);
        StringBuilder secondNumber = convertFromSymbolsToNumericChars(secondStr);
        compareAndPrint(firstNumber, secondNumber);
    }
    static StringBuilder convertFromSymbolsToNumericChars(String strNumber) {
        StringBuilder resultStr = new StringBuilder();
        int i = 0;
        while (i < strNumber.length()) {
            if (strNumber.charAt(i) == 'o') {
                resultStr.append('1');
                i += 3;
            } else if (strNumber.charAt(i) == 'z') {
                resultStr.append('0');
                i += 4;
            }
        }
        return resultStr;
    }

    static void compareAndPrint(StringBuilder firstNumber, StringBuilder secondNumber) {
        int f = 0;
        int s = 0;
        while (f < firstNumber.length() && s < secondNumber.length()) {
            f = getPosOfOne(firstNumber, f);
            s = getPosOfOne(secondNumber, s);
            if (firstNumber.length() - f > secondNumber.length() - s) {
```

Submit

Next