☆ Balanced Or Not?

Consider a string, *expression* consisting of the characters `<` and `>` only. We consider the string to be *balanced* if each `<` always appears *before* (i.e., to the left of) a corresponding `>` character (they do not need to be adjacent). Moreover, each `<` and `>` act as a unique pair of symbols and neither symbol can be considered as part of any other pair of symbols. For example, the strings `<<>>`, `<>`, and `<><>` are all *balanced*, but the strings `>>`, `<<>`, and `>><<` are *unbalanced*.

To balance a string, we can replace any `>` character with `<>` at most *maxReplacement* times. Given an *expression* and the value of *maxReplacement*, can you turn an unbalanced string into a balanced one?

Complete the *balancedOrNot* function in the editor below. It has the following parameters:

1. An array of *n* strings, *expressions*, denoting the list of expressions to check.
2. An array of *n* integers, *maxReplacements*, where $maxReplacements_i$ denotes the maximum number of replacements allowed when attempting to balance $expressions_i$.

The function must return an array of integers where each index i ($0 \le i < n$) contains a *1* if $expressions_i$ is balanced or a *0* if it is not.

Input Format

The first line contains an integer, *n*, denoting the size of *expressions*.
Each line *i* of the n subsequent lines (where $0 \le i < n$) contains a string describing $expressions_i$.
The next line contains an integer, *m*, denoting the size of *maxReplacements*.
Each line *i* of the n subsequent lines (where $0 \le i < n$) contains a string describing $maxReplacements_i$.

Constraints

- $1 \le n \le 10^2$
- $1 \le length\ of\ expressions_i \le 10^5$
- $0 \le maxReplacements_i \le 10^5$

Output Format

The function must return an array of integers where each index i ($0 \le i < n$) contains a *1* if $expressions_i$ is balanced or a *0* if it is not.

Sample Input 0

```
2
2
2
```

Sample Output 0

```
1
0
```

Explanation 0

We process *expressions* = *["<>>>", "<>>>>"]* and *maxReplacements* = *[2, 2]* like so:

0. For string `<>>>` with $maxReplacements_0 = 2$, it becomes balanced after two replacements: `<>>>` → `<><>>` → `<><><>`. Because the string was converted in ≤ $maxReplacements_0$ replacements, we store a *1* in index *0* of our return array.
1. For string `<>>>>` with $maxReplacements_1 = 2$, becomes balanced after three replacements: `<>>>>` → `<><>>>` → `<><><>>` → `<><><><>`. Because the string was converted in > $maxReplacements_1$ replacements, we store a *0* in index *1* of our return array.

We then return the array *[1, 0]* as our answer.

Sample Input 1

```
2
<>
<>><
2
1
0
```

Sample Output 1

```
1
0
```

Explanation 1

We process *expressions* = *["<>", "<>><"]* and *maxReplacements* = *[1, 0]* like so:

0. For string `<>` with $maxReplacements_0 = 1$, it is already balanced and needs no replacements. Because the string is balanced in ≤ $maxReplacements_0$ replacements,

performing *0* replacements), so we store a *0* in index *1* of our return array.

We then return the array *[1, 0]* as our answer.

YOUR ANSWER

We recommend you take a quick tour of our editor before you proceed.
The timer will pause up to 90 seconds for the tour. Start tour

Draft saved 04:33 pm | Original code | Java 7

```
1 ▸ import ↔;
6
7   public class Solution {
8
9 ▾ /*
10   * Complete the function below.
11   */
12
13 ▾     static int[] balancedOrNot(String[] expressions, int[]
     maxReplacements) {
14
15
16      }
17
18
19 ▸     public static void main(String[] args) throws IOException{↔}
56  }
```

Line: 12 Col: 1

Run Code | Submit code & Continue

(You can submit any number of times)

Test against custom input

5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23