

 \equiv

☆ Balanced Or Not?

3

5

6

7

9

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

Sample Input 0

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.

0	
0	
0	

2

8

5

6

7

8

9

10

11

12

13

14

15

16

17



19

20

21

22

23

2 2

Sample Output 0

```
1
0
```

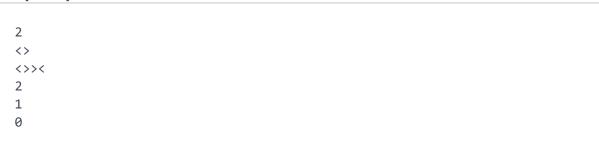
Explanation 0

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

- 0. For string $\langle \rangle \rangle$ with maxReplacements₀ = 2, it becomes balanced after two replacements: $\langle \rangle \rangle \rightarrow \langle \rangle \langle \rangle \rightarrow \langle \rangle \langle \rangle$. Because the string was converted in \leq maxReplacements₀ replacements, we store a 1 in index 0 of our return array.
- 1. For string $\langle \rangle \rangle \rangle$ with $maxReplacements_1 = 2$, becomes balanced after three replacements: $\langle \rangle \rangle \rangle \rightarrow \langle \rangle \langle \rangle \rangle \rightarrow \langle \rangle \langle \rangle \langle \rangle \rangle$. Because the string was converted in > maxReplacements₁ 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



Sample Output 1

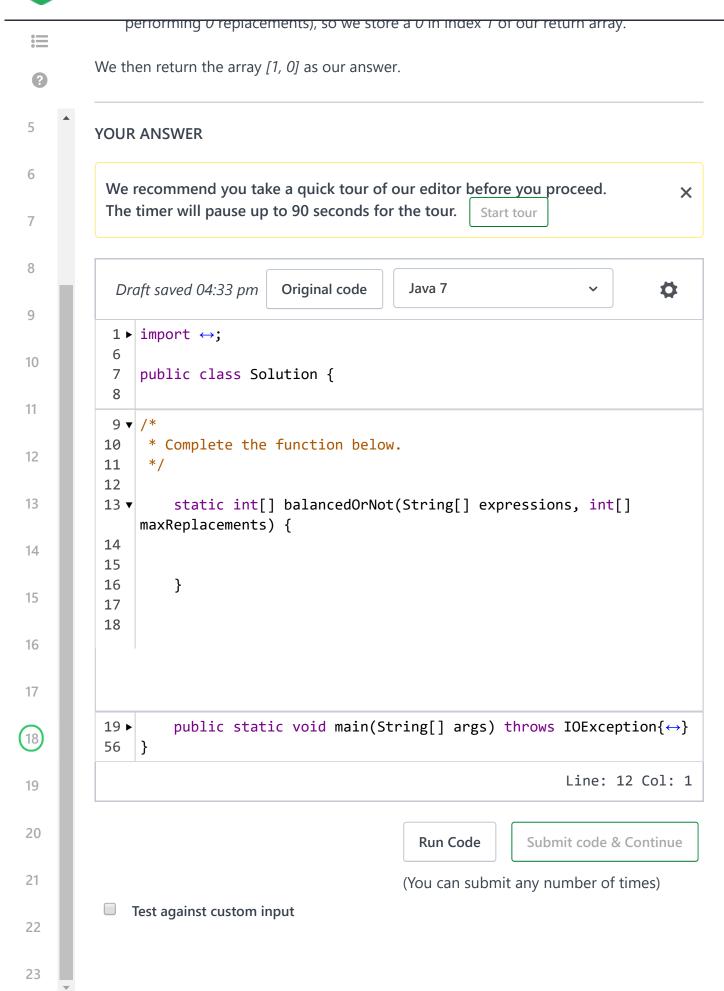
1		
0		

Explanation 1

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

0. For string $\langle \rangle$ with maxReplacements₀ = 1, it is already balanced and needs no replacements. Because the string is balanced in $\leq maxReplacements_0$ replacements,







 \equiv

?

About Privacy Policy Terms of Service