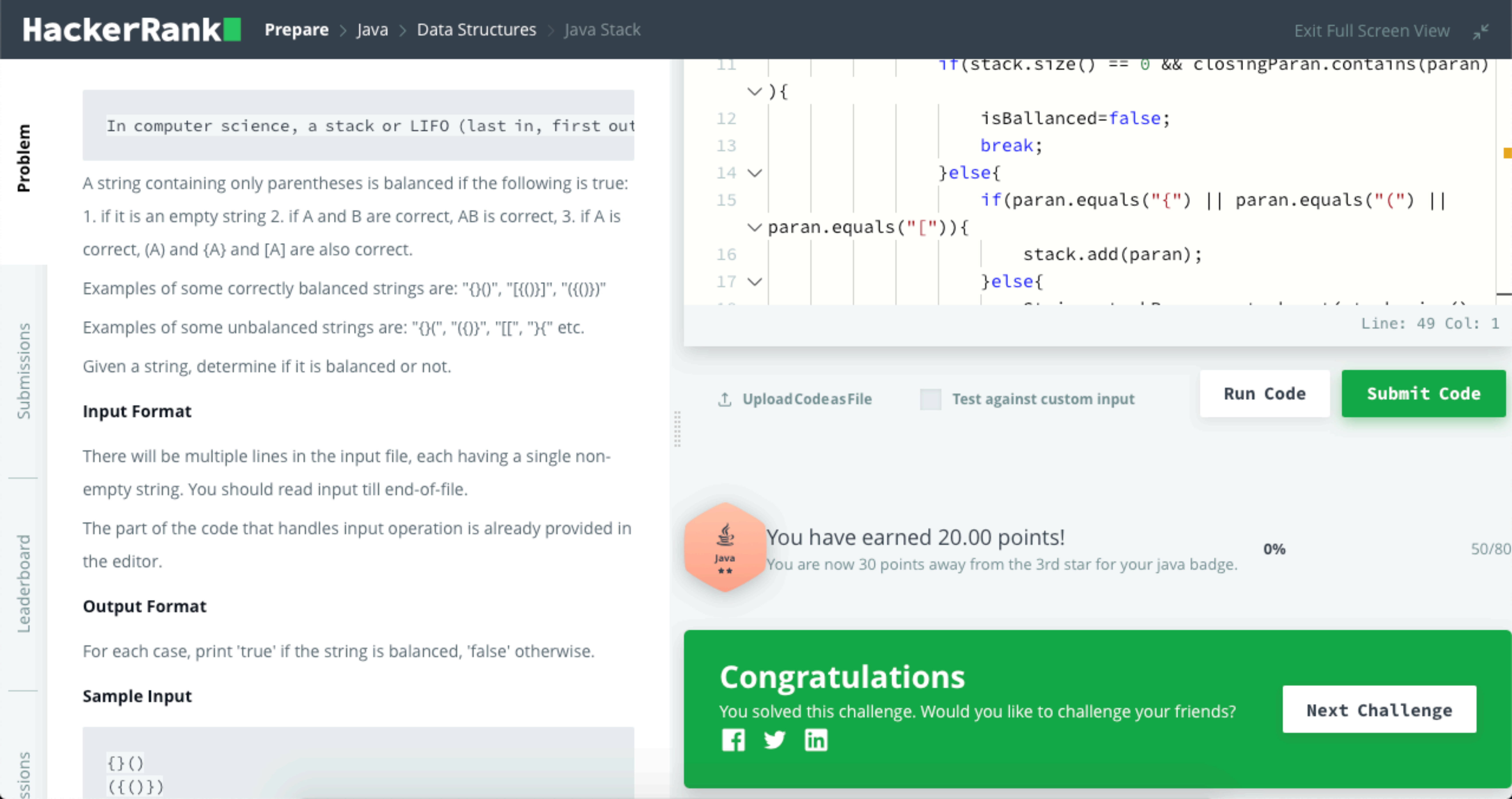

```
In computer science, a stack or LIFO (last in, first out
```

A string containing only parentheses is balanced if the following is true: 1. if it is an empty string 2. if A and B are correct, AB is correct, 3. if A is correct, (A) and {A} and [A] are also correct.

Examples of some correctly balanced strings are: "{}()", "[{()}]", "({()})"

Examples of some unbalanced strings are: "{}(", "({)}", "[[", "}{" etc.

Given a string, determine if it is balanced or not.

**Input Format**

There will be multiple lines in the input file, each having a single non-empty string. You should read input till end-of-file.

The part of the code that handles input operation is already provided in the editor.

**Output Format**

For each case, print 'true' if the string is balanced, 'false' otherwise.

**Sample Input**

```
{}()
({()})
```

```
                if(stack.size() == 0 && closingParan.contains(paran)
){
                    isBallanced=false;
                    break;
                }else{
                    if(paran.equals("{") || paran.equals("(") ||
paran.equals("[")){
                        stack.add(paran);
                    }else{
```

Line: 49 Col: 1

You have earned 20.00 points!

You are now 30 points away from the 3rd star for your java badge.

## Congratulations

You solved this challenge. Would you like to challenge your friends?