https://course.acciojob.com/idle?question=e16170b9-480d-4bff-be85-dacd2afc2e48

- EASY
- Max Score: 30 Points

Balanced Expression

You are given a string `exp` representing an expression. You are required to check if the expression is balanced i.e. closing brackets and opening brackets match up well.

The string `exp` contains the characters '(', ')', '{', '}', '[', ']', operators i.e. '-', '+', '*', '/' and lowercase english alphabets.

Example:

```
[(a+b)+{(c+d)*(e/f)}] -> true
[(a+b)+{(c+d)*(e/f)]} -> false
[(a+b)+{(c+d)*(e/f)} -> false
([(a+b)+{(c+d)*(e/f)}] -> false
```

An input string is Balanced if:

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.

Your task is to complete the function `expBalanced` which receives the input expression `exp` as parameter and returns true or false depending on if the expression is balanced or not.

Input Format

The only line contains string `exp`

Output Format

Print `true` is the expression is balanced otherwise print `false`.

Example 1

Input

```
[(a+b)+{(c+d)*(e/f)}]
```

Output

```
true
```

Explanation As `[` **was closed with `]`, `(` was closed with `)` and `{` was closed with `}` therefore output is`true`.**

Example 2

Input

```
[(a+b)+{(c+d)*(e/f)]}
```

Output

```
false
```

Explanation The bracket `{` **and `[` was not closed, Therefore output is `false`.**

Constraints

`1 <= exp.length <= 10^4`

Topic Tags

- Stacks

My code

```
// n java
import java.io.*;
import java.util.*;

class Solution{
    boolean expBalanced(String str){
        // write code here
                  Stack<Character>st=new Stack<>();
                for(int i=0;i<str.length();i++)
                        {
                                if(str.charAt(i)=='(' ||str.charAt(i)=='{'
||str.charAt(i)=='[')
                                        st.push(str.charAt(i));
                                                //if(str.charAt(i)==')'
||str.charAt(i)=='}' ||str.charAt(i)==']')
                                             if(str.charAt(i)==')')
                                                {
                                                        if(!st.isEmpty() &&
st.peek()=='(')
                                                                st.pop();
```

```
                                            else
                                            {
                                                return false;
                                                 //  return;
                                            }
                                                 }

                                 if(str.charAt(i)=='}')
                                                 {
                                                         if(!st.isEmpty()
&&st.peek()=='{')
                                                                st.pop();

                                            else
                                            {
                                                return false;
                                                  // return;
                                            }
                                                 }

                     if(str.charAt(i)==']')
                                                 {
                                                         if(!st.isEmpty() &&
st.peek()=='[')
                                                                st.pop();

                                            else
                                            {
                                                return false;
```

```
                                                  //  return;
                                            }
                                                 }




                        }
                if(st.empty())
                        return true;
                  return false;
    }
}

public class Main {

    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
        String str = br.readLine();
        Solution Obj = new Solution();
        System.out.println(Obj.expBalanced(str));
    }
}
```