1 Assignment 8 Introduction

This assignment is due *before* 11:59pm on the listed date. Which means, submissions made *on or after* 11:59pm will be counted as a late submissions. Late submissions *before* 11:59pm on the following day will receive a 25% point deduction as penalty. Submissions made *on or after* 11:59pm on the same day will *not* be graded. We strongly recommend completing the assignment before then.

It is imperative that you meticulously follow the submission process outlined at the end of this assignment. Incorrectly structured submissions will receive a 10% point deduction as a penalty.

Assignments due on Mondays generally involve materials covered in lectures from the previous week, whereas assignments due on Thursdays involve materials covered in lectures from the running week. So be sure to watch the lectures and go over the reading materials before attempting the assignments.

Good luck!

2 Balancing Act (40 points)

As you already know, in the syntax of `C++` '`()`' (parentheses), '{}' (curly braces), and '`[]`' (brackets) appear in nested pairs. In a program with correct syntax, these operators will be nested and matched (i.e. for every opening parenthesis, curly brace or bracket, there will be a closing parenthesis, curly brace or bracket). To determine whether this condition holds for a particular program, you can ignore all the other characters and simply look at the pattern formed by the parentheses, curly braces, and brackets. In a balanced configuration, all the operators match up correctly, as shown in the following example:

```
{ ( [ ] ) ( [ ( ) ] ) }
```

The following configurations however, are not balanced, due to their stated reasons:

```
( ( [ ] ) // Missing a closing parenthesis
) (       // Closing parenthesis appears before the opening parenthesis
{ ( } )   // Parentheses and braces are improperly nested
```

Your task is to write a program using the function `IsBalanced`, that takes in a configuration as input, and returns `true` if it is balanced, otherwise `false`. You may refer to the following as a starting point:

```
#include <iostream>
#include <string>

bool IsBalanced(const std::string& config) {
    //Your code here
}

int main() {
    std::string config{""};
    std::cin >> config;
    std::cout << IsBalanced(config);
    return 0;
}
```

Some test cases are provided below:

Input:

```
( { [ ] } ) [ ] { ( ) }
```

Output:

```
1
```

Input:

```
}
```

Output:

```
0
```

Input:

```
( [ ] ) { [ ( ) ] }
```

Output:

```
1
```

Input:

```
] [
```

Output:

```
0
```

Input:

```
( { ) }
```

Output:

```
0
```

General Hints and Instructions:

- You are not allowed to change the function declaration.
- Your first instinct may be to compare the number of opening and closing operators, and checking if they are equal. It should be obvious why that approach alone will not yield a viable solution.
- This is a string manipulation problem. Consider making heavy use of the `find`, `substr`, and `replace` functions.
- Your program should compile and run. You can assume the input will always be a string of length one or greater, consisting of only parentheses, curly braces, and brackets. There will not be any other characters (including whitespaces) in the input string.
- Do not add any unnecessary components to your solution. The problem statement is not asking for any header files or additional input/output prompts.

3 Balancing Act II (60 points)

Your task is to write a program using the **recursive** function `IsBalancedRec`, that takes in a configuration as input, and returns `True` if it is balanced, `False` otherwise. You may refer to the following as a starting point:

```
#include <iostream>
#include <string>

bool IsBalancedRec(const std::string& config) {
    //Your code here
}

int main() {
    std::string config{""};
    std::cin >> config;
    std::cout << IsBalancedRec(config);
    return 0;
}
```

While there are multiple ways to design the recursive process, a brief description of one of the recursive processes is given below:

- Receive string `config` in `IsBalancedRec` function.
- Is `config` empty?
  - If yes, then `config` is balanced.
- Does `config` have a pair of '`()`' (parentheses), '{}' (curly braces), or '`[]`' (brackets)?
  - If yes, then remove the pair from `config` and call `IsBalancedRec` with the updated `config`.
  - If no, then `config` is not balanced.

Note that in the recursive process above, `config` is updated even though in the function parameter `config` is declared as a const reference to a string, which you are not allowed to change. At this point in the course, you should be able to figure out how to get around this issue.

Some test cases are provided below:

Input:

```
( { [ ] } ) [ ] { ( ) }
```

Output:

```
1
```

Input:

```
}
```

Output:

```
0
```

Input:

```
( [ ] ) { [ ( ) ] }
```

Output:

```
1
```

Input:

```
] [
```

Output:

```
0
```

Input:

```
( { ) }
```

Output:

```
0
```

General Hints and Instructions:

- You are not allowed to change the function declaration.
- Your solution must use recursion. Solutions using while loops, for loops, etc will not be accepted.
- This is a string manipulation problem. Consider making heavy use of the `find`, `substr`, and `replace` functions.
- Your program should compile and run. You can assume the input will always be a string of length one or greater, consisting of only parentheses, curly braces, and brackets. There will not be any other characters (including whitespaces) in the input string.
- Do not add any unnecessary components to your solution. The problem statement is not asking for any header files or additional input/output prompts.

4 Assignment 8 Submission Process

- Create a folder, name it `your_msu_id8`. For example, if your MSU email is `johndoe@msu.edu`, then you should name the folder `johndoe8`.
- For each programming task, create a sub-folder inside your `your_msu_id8` folder, and name it as the number that corresponds to the programming task number. For this assignment, there should be two sub-folders named '2' and '3'.
- Inside each sub-folder, put the `main.cpp` for the appropriate solution.
- Compress/Zip `your_msu_id8` folder and name it `your_msu_id8.zip`. For example, if the name of your folder is `johndoe8`, then you need to create a zip file named `johndoe8.zip`. Zip file guide: `https://copyrightservice.co.uk/reg/creating-zip-files`.
- Submit the zip file through D2L.