CS 280
Fall 2022
Recitation Assignment 7

November 10, 2022

Due Date: Monday, November 14, 2022, 23:59
Total Points: 6

Write a C++ function that uses a stack to check for the balancing of three types of delimiter symbols: Braces {}, parentheses (), and square brackets []. The function ignores all scanned characters that are not one of the three types of delimiter symbols. The function pushes each scanned opening delimiter into the stack. While for each scanned closing delimiter, the function checks whether the scanned closing delimiter is matched with the opening delimiter at the top of the stack. If there is a matching between the two symbols the top of stack is popped and the scanning process of the string continues. However, four cases are identified for terminating the execution of the function. Your function should print out the corresponding message associated with the function termination case. The four cases are:

Case 1: The scanning of the string is completed and the stack is empty. This situation indicates the string is containing balanced delimiters. The printed message should be:
"Stack is empty and the scanning of input string is completed."

Case 2: The scanning of the string is completed and the stack is not empty. This situation indicates the string does not contain balanced delimiters. The printed message should be:
"Stack is not empty and the scanning of input string is completed."

Case 3: The scanning of the string cannot be completed due to mismatching, and the stack is not empty. This situation indicates the string does not contain balanced delimiters. The printed message should be:
"Stack is not empty and the scanning of input string is not completed."

Case 4: The scanning of the string cannot be completed because the stack has become empty. This situation indicates the string does not contain balanced delimiters. The printed message should be:
"Stack is empty and the scanning of input string is not completed."

For example, the following strings do not contain balanced delimiter symbols:

“abc{de{f[()}}”,

“abc{de(f[g])}}”, and

“abc{de[fg(h)i}”.

While the following strings contain balanced delimiter symbols:

```
“abc{de[f]g(h)i}”
```

```
“we rt   uof  asd”
```

The function has the following header definition:

```
bool BalancedDelimiters(string & inputString);
```

Where, the function accepts a reference to an input string parameter and returns true if the `inputString` argument contains balanced delimiter symbols, and false if it does not.

Vocareum Automatic Grading

- A driver program is provided for testing the implementation, called “ra7prog.cpp”, on Vocareum. The “ra7prog.cpp” will be propagated to your Work directory. The program reads from a file name specified in the command line as an argument. The driver program handles the opening of the file for reading and checks if the file is empty or not.
- You are provided by a set of 5 testing files associated with Recitation Assignment 7. Vocareum automatic grading will be based on these testing files. You may use them to check and test your implementation. These are available in a compressed archive “RA7 Test Cases.zip” on Canvas assignment. The testing case of each file is defined in the Grading table below.
- “ra7prog.cpp” is available with the other assignment material on Canvas.

Submission Guidelines

- Please upload your implementation to Vocareum as a “balancedDelimiters.cpp” file. The file should include the implementation of the function `BalancedDelimiters()`.

- **Submissions after the due date are accepted with a fixed penalty of 25% from the student’s score. No submission is accepted after Wednesday 11:59 pm, November 16, 2022.**

Grading Table

Item	Points
Compiles Successfully	1
String 1	1
String 2	1
String 3	1
String 4	1
String 5	1
Total	6