Programming Exam Choice 13

Start by downloading the question 13 files from the moodle into a new directory on your machine. You will be changing the code in the question13.cpp file. You have a Makefile to handle building this code.

Part 1. 5 points

You will use the array_stack we made in class to process a string of brackets, like “{ { } } { }”, to decide if they are balanced. To do this, you’ll write the function below to return true if the brackets and parentheses in the string are balanced. For the test, I rewrote the stack to accept characters, not integers, so you can just push characters from the string onto the stack.

```
bool balanced_brackets(const string&s)
{

}
```

When your code is working right, it should be able to tell that

{ }	is balanced
{ { } } { }	is balanced
{ { }	is not balanced

Now that you have mixed brackets, you need to make sure that the brackets match as you pop them off the stack. For example,

{ ( ) ( ) [( )] }	is balanced
{ ( ) ( ) [( ]) }	is not balanced
{ [( ) ( )] [( )] } ( )	is balanced

Test your code on a few of the above examples and when you are satisfied, please upload your question13.cpp file to the moodle. Your TA may ask you to zip other files in as well.

Logic of problem laid out in comments:	50%
Code compiles with no errors or warnings:	10%
Code has no run time errors:	10%
Code gives correct answers for all inputs:	20%
Code is clean and easy to read:	10%