

## Extra Assignment Tests

**Test 1:** A simple test implementing plain text with closed-off brackets to see if text works well with the checked characters. The brackets turn out to be balanced.

```
ENTER CHARACTERS:  
HELLO WORLD () [] {}  
THE BRACKETS ARE BALANCED!
```

**Test 2:** A test implementing a standard math equation but with the last bracket missing. This proves that the program also checks for any remaining brackets left in the stack once the whole string is checked.

```
ENTER CHARACTERS:
((2x+y) - 4(x-y)
THE BRACKETS ARE NOT BALANCED!
```

**Test 3:** A series of coordinate values with each set being closed off to test if spacing between properly closed brackets works.

```
ENTER CHARACTERS:  
{4,6},{7,3},{9,4}  
THE BRACKETS ARE BALANCED!
```

**Test 4:** No text was entered for this test. The statement ends up being true because the counterargument against this result would be that the brackets (which don't exist) are NOT balanced, which is simply not true. Any formula or text written would still be properly read, therefore the result is balanced.

```
ENTER CHARACTERS:
THE BRACKETS ARE BALANCED!
```

**Test 5:** A long series of random brackets were entered to check if the program can handle a large string input and still produce the correct result. In this case the brackets are clearly not balanced, and the result is correct.

```
ENTER CHARACTERS:
}{({})}({)((()))(())}{{}{}{[]]]]]][[[[[[[[[[
THE BRACKETS ARE NOT BALANCED!
```