|  |  |
| --- | --- |
| **Project Name: Aegis Project 1: Voting System Team#7** | |
| **Test Stage: Unit X System \_** | **Test Date:** 3/27/21 |
| **Test Case ID#:** Helper\_5: popItem | **Name(s) of Testers:** Michael/Donald/Lucky/Grant |
| **Test Description:**  Parses and pops a string that is in between the two char arguments. |  |
| **Automated: yes X no \_\_\_** | **Indicate where you are storing the tests (what file) and the name of the method/functions being used.**  Stored in file:  Method names:   * popItem(string, char, char) |
| **Results: Pass \_\_X\_\_\_** Fail **\_\_\_\_\_\_\_\_** |  |
|  |  |
| **Preconditions for Test:**  The strings and int are set up correct as well as any other variables. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
|  |  |  |  |  |  |
| 1 | Create a string and symbols to test on | std::string line = "[The quick brown fox] jumped[] over [the lazy dog]";  char l = '[', r = ']'; |  |  |  |
| 2 | Test first instance on line and check that it pops the string from the original string (*line*) |  | EXPECT\_EQ("The quick brown fox", popItem(line, l, r)) = true  EXPECT\_EQ(" jumped[] over [the lazy dog]", line) = true | EXPECT\_EQ("The quick brown fox", popItem(line, l, r)) = true  EXPECT\_EQ(" jumped[] over [the lazy dog]", line) = true |  |
| 3 | Test second instance on the new *line* that’s been popped from in step 2 |  | EXPECT\_EQ("", popItem(line, l, r)) = true  EXPECT\_EQ(" over [the lazy dog]", line) = true | EXPECT\_EQ("", popItem(line, l, r)) = true  EXPECT\_EQ(" over [the lazy dog]", line) = true |  |
| 4 | Test third instance on the new *line* that’s been popped from in step 3 |  | EXPECT\_EQ("the lazy dog", popItem(line, l, r)) = true  EXPECT\_EQ("", line) = true | EXPECT\_EQ("the lazy dog", popItem(line, l, r)) = true  EXPECT\_EQ("", line) = true |  |

**Post condition(s) for Test:** 

The value returned should be the substring that was popped from the *line.*