**Submission of Your Work**

You need to prepare and submit ONE SINGLE MS Word document to Canvas (in your lab section) as LastName\_FirstName\_proj3.docx. It must contain:

* Your NAME only on page 1
* Source code. Copy/Paste your final source code. You must include standard “comment header” even if code is provided. *Do Not* paste a snippet of your source code, it must be copy/pasted.
* Initial test plan. After reading the project requirements, but **before** beginning to code, create the test case table, below, completed through column Expected Output. Include in your report *AFTER* source code.
* Final test plan. Write your program then complete the **test table** with actual output results and include in your report *AFTER* your source code.
* Output results. Paste in a snippet of output showing results for **every listed test case in your final test plan**, labeled with test case #

Test Table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Expected Output | Actual Output | Test Pass / Fail |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |

* Add / delete rows from Test Table as necessary
* Modify column widths as necessary
* Test both valid and invalid input
* Test for every output expected
* If failure is an expected output and it happens then that test Passes
* Any test that fails means the program must be fixed so that it passes the test
  + Failing tests need a new test row, ie 1a, 1b, etc, showing corrections from original

The goal of this program is creating a circular doubly linked list. This means it has two links, one going forward and one going backward, and that when initially created (first insertion), nextPtr points to itself and prevPtr points to itself [there are no nullptrs once the list is created].

Use this structure:

struct node {  
 char letr;  
 node \* nextPtr,  
 \* prevPtr;  
 };

Write functions to:

* insert new characters in sort order
* find any value. Search forward or backward (whichever is closer; can you determine this? Use a best guess approach to determine the direction to go). You MUST use search function as part of your insert and delete processes. Search MUST be recursive.
* print the list either forward or backward, as requested
* delete any value

Use exception handling (try, throw, catch) when attempting to delete an item when the list is empty, when the value isn’t in the list, or printing an empty list.

Successfully demonstrate all functionality.