**Assignment 2**

**Due, Tuesday, March 3, 2015 up to 100**

**Wednesday, March 4, 2015 up to 90**

**Thursday, March 5, 2015 up to 80**

**Friday, March 6, 2015 up to 70**

**Beyond 3/6/2015, beg for mercy from Professor Whiting!**

**Deliverables**

To complete this assignment you must submit to Webcourses:

1. source code .c file

**Introduction**

This assignment provides more practical experience parsing data files and creating a linked list and builds upon it with recursion and writing a data file.

**References**

1. Text book, Chapter 3. Chapter 7.
2. Source code examples on Webcourses, LinkedList.c, Recursion.c, RecursiveReverse.c TowerOfHanoi.c
3. Online tutorials:
   1. <http://www.cprogramming.com/tutorial/c-tutorial.html>
   2. <http://fresh2refresh.com/>
   3. <http://www.tutorialspoint.com/c_standard_library/>

**ATTENTION MAC USERS!**

If you are using a Mac operating system, I understand that you have to fully qualify the path to the files for the program to run. I ask that before you submit your assignment please remove the fully qualified path so that the **fopen("AssignmentTwoInput.txt", "r");** function call has only the file name. Thank you!

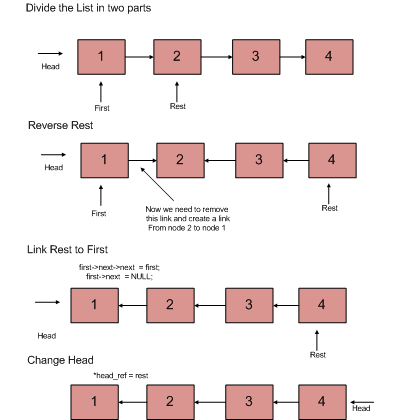
**Tasks and Rubric**

|  |  |  |
| --- | --- | --- |
| Activity | | Points |
| Define a linked list node to include | char produce[20];  char type[20];  char soldBy[20];  float price;  int quantityInStock;  struct produceItem \*next; | 5 |
|  | **NO GLOBAL HEAD ALLOWED!** | 5 |
| main() | - Provide the user a menu of the following options:  1. Stock Produce Department  // call function to read in the data file  2. Display Produce Inventory  // call function to display the data in the format shown in figure  3. Reverse Order of Produce Inventory  // call function to recursively reverse the order of the linked list  4. Export Produce Inventory  // call function to write out the ending inventory in the format shown in figure  5. Exit Program  // exit the program  - Use a conditional statement to evaluate the user’s selection  -See Figure 1 for the output display | 5 |
| Function to read the data file as nodes of a linked list | Read in the contents of data file "AssignmentTwoInput.txt" into the node data structure and generate a linked list using stack methodology of pushing each new node on to the linked list | 5 |
| Test Case 1 | Perform Test Case 1, results in expected outcome | 5 |
| Function to display the linked list | Traverse the linked list and display the data of each node formatted as shown in Figure 3 | 5 |
| Test Case 2 | Perform Test Case 2, results in expected outcome | 5 |
| Function to reverse the linked list | Use recursion to reverse the order of the linked list | 10 |
| Test Case 3 | Perform Test Case 3, results in expected outcome | 5 |
| Test Case 4 | Perform Test Case 4, results in expected outcome | 5 |
| Test Case 5 | Perform Test Case 5, results in expected outcome | 5 |
| Test Case 6 | Perform Test Case 6, results in expected outcome | 5 |
| Function to write produce inventory to a file | Write out the elements of the linked list to data file “AssignmentTwoOutput.txt” so that it replicates the example provided | 5 |
| Test Case 7 | Perform Test Case 7, results in expected outcome | 5 |
| Exit the program | Write the appropriate code to exit the program | 5 |
| Test Case 8 | Perform Test Case 8, results in expected outcome | 5 |
| Compile | Source compiles with no errors | 5 |
| Run | Source runs with no errors | 5 |
| Comments | Source includes comments | 5 |
| Total |  | **100** |

**Perform the following test cases**

|  |  |  |
| --- | --- | --- |
| Test Cases | | |
|  | **Action** | **Expected outcome** |
| Test case 1 | Select option 1 | Data file AssignmentTwoInput.txt is read in, string tokenized, and a linked list is generated using stack logic so that each new node is pushed on to the linked list |
| Test case 2 | Select option 2 | Display the elements of the linked list, see Figure 3 |
| Test case 3 | Select option 3 | Reorder the linked list from end to beginning |
| Test case 4 | Select option 2 | Display the elements of the linked list, see Figure 5 |
| Test case 5 | Select option 3 | Reorder the linked list from end to beginning |
| Test case 6 | Select option 2 | Display the elements of the linked list, see Figure 2 |
| Test case 7 | Select option 4 | Data file AssignmentTwoOutput.txt is written, see Figure 6 |
| Test case 8 | Select option 5 | Program exits and quits |

**Recursive Method:**



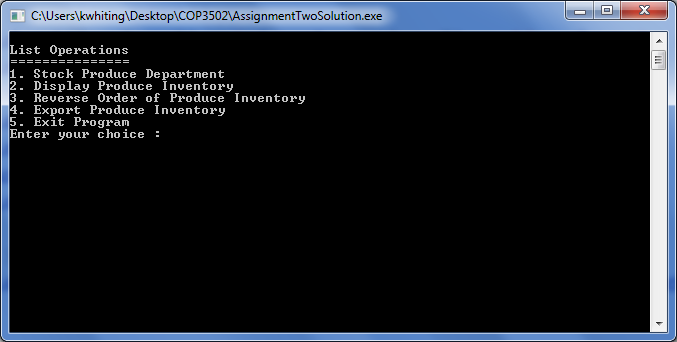


Figure 1 Application Start

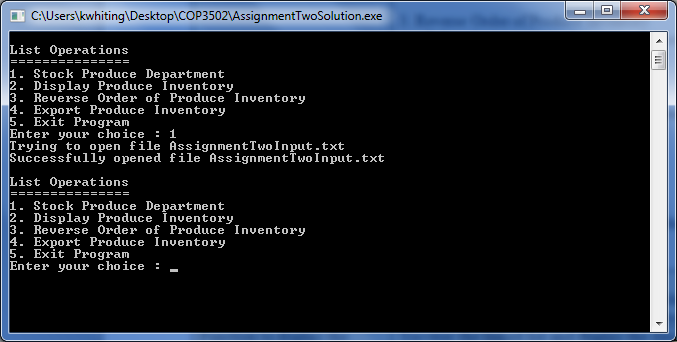


Figure 2 After Option 1

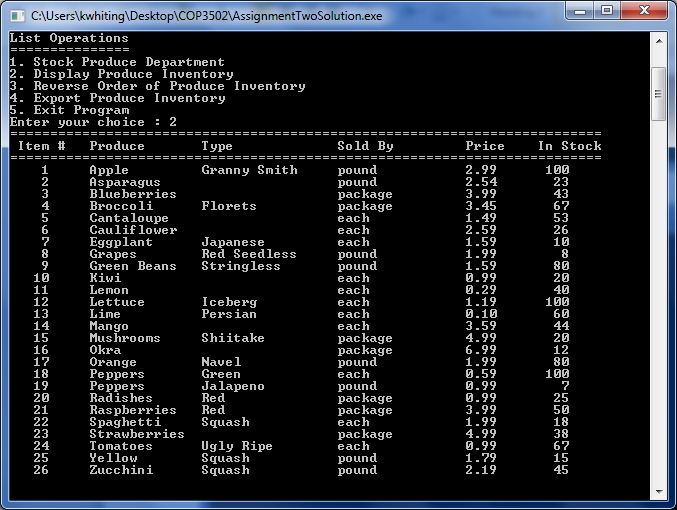


Figure 3 After Option 2

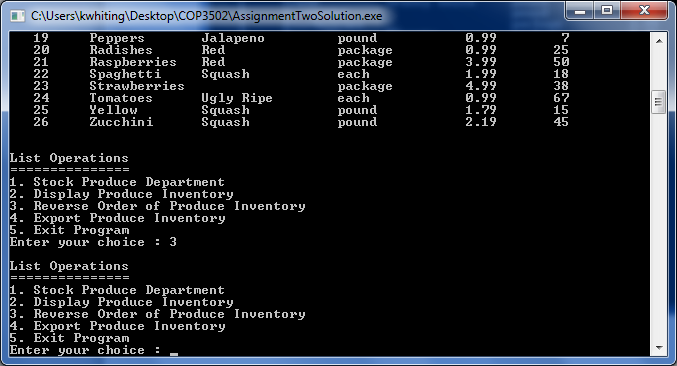


Figure 4 After Option 3

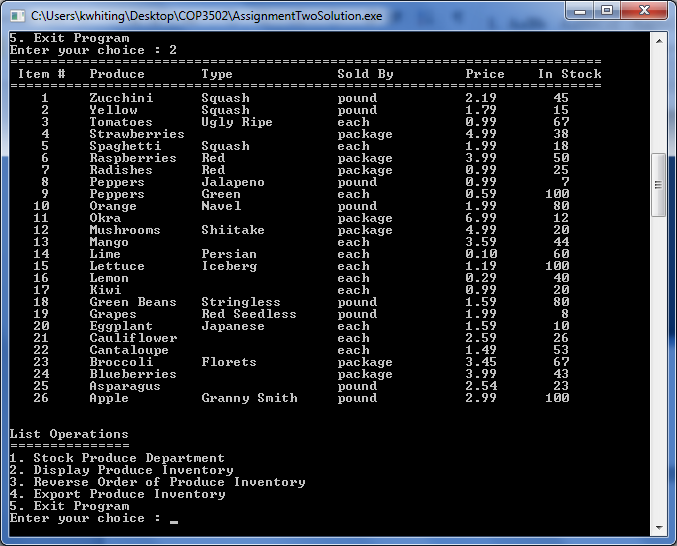


Figure 5 After Option 2

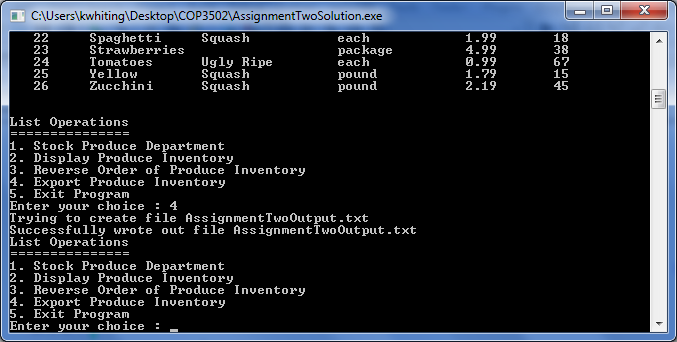


Figure 6 After Option 4