Programming Assignment 3 - Linked List

Due Date : Section 0 - Monday March 5th , 2018 - No Later than 2:15 pm

Section 1 - Monday March 5th , 2018 - No Later than 3:45 pm.

Section 2 - Monday March 5th , 2018 - No Later than 5:15 pm.

For this assignment you will implement linked List ADT using actual pointers. The purpose of the program is to :

A. Build 2 lists (list_1 and list_2). Each list consist of 15 random integer numbers that are between 0 and 20 exclusive Display List_1 and List_2.
B. Create new list (list_3) that contain all elements of list_1 followed by all elements of list_2. Display list_3
C. Display the number of occurrences of the first element in list_3.
D. Create a new list (list_4) by removing all duplicate numbers that are in list_3 . Display List_4.
E. Display number of elements in list_4.
F. Display list_4 backward.
G. Move the last element in list_4 and places it in after the 3rd element in the same list. Display List_4
H. Sort list_4 in increasing order. Display the sorted list_4.

Repeat Process A – H until the user terminates the program.

Notes :

- You design your own output format. However, The output must be easy to follow.
- Must place the following after before terminating the program.
- Validation must be done with the repeat process..

This LL program is implemented by :
Husain Gholoom - March 5 th, 2018

Style Guidelines:

At the beginning of your program (and **before** the #include statement), include the following :

Header comments (file documentation block) should be at the top of each file and should contain: Author / s, Due Date, Assignment Number, Course number and section, Instructor, and a brief description of the purpose of the code in the file. For example :

```
//      Roster Number / s :      xxxxxxxxx
//
//      Author / s : (Your name here!!)
//
//      Programming Assignment Number 3
//
//      Spring 2018 - CS 3358 - Your Section Number
//
//      Due Date :
//
//      Instructor:  Husain Gholoom.
//
//       <Brief description of the purpose of the program>
```

Variable names :

- Must be meaningful.
- The initial letter should be lowercase, following words should be capitalized, no other caps or punctuation (i.e. `weightInPounds`).
- Each variable must be declared on a separate line with a descriptive comment.

Named constants :

- Use for most numeric literals.
- All capitals with underscores (i.e. `TX_STATE_SALES_TAX`)
- Should occur at top of function, or global (only if necessary)

Line length of source code should be no longer than 80 characters (no wrapping of lines).

Indentation :

- Use 2-4 spaces (but be consistent throughout your program).
- Indent blocks, within blocks, etc.
- Use blank lines to separate sections.

Comments for variables :

All variable definitions should be commented as follows:

```
int  gender;    // integer value for the gender,
                // 1 = Male , 2 = Female ,
```

Rules :

1. Your program **must compile** and run using latest version of codeblocks under windows.
2. The entire program must be **documented according the style above . See the website for the sample programming style program.**
3. You must use the appropriate libraries in writing this program, however, you are **not allowed** to use libraries that support the functionality of linked lists such as list or forward_list. . You are also not allowed to use recursion.
4. Your program must use a class along with at appropriate functions when applicable.
5. You must name your program as :

   - **LastName_FirstName_3358_0_PG3_LL.cpp (not .cbp) or**
   - **LastName_FirstName_3358_1_PG3_LL.cpp (not .cbp) or**
   - **LastName_FirstName_3358_2_PG3_LL.cpp (not .cbp) or**

   Where LastName is your Last Name and FirstName is your First Name.

6. Every one must upload the electronic version of the program no later than the starting of class time on the due date. **No late assignments will be accepted.** **DO NOT** send your assignment solution via email. **Group members must upload identical copy of the assignment.**

   Use TRACS To upload electronic version of your program

7. You must **also** turn in hard copy of your source code no later than the starting of class time on the due date . should the hard copy consist of more than one page , then , the hard copy must be **stapled**. if you are unable to turn in a printout during class, you can take the program to the computer science department and hand it to the front desk personal (Comal 211) before the deadline. Make sure that the front office stamps the program. Make sure that include the date and time. Finally ,make sure that they place the program in my mailbox. **Only one copy per group.**

   DO NOT slide your program under my office door – It will **NOT** be accepted

The following points will be deducted if :

- Incorrect file format such as uploading .cbp instead of .cpp , missing electronic copy , missing the hardcopy using .h and .cpp files , compilation Errors , Using arrays or global arrays , vectors or global vectors , not using class , using recursion … etc **(- 10 points)**

- Other **(at least 1 point each)** :

  - Logical Errors
  - Unable to read the source code due to unclear printing
  - Incorrect program file name.
  - More than one hardcopy per group or Hard copy is not stapled.
  - Incorrect Style **such as but not limited to** missing comments or program documentations , missing roster number , missing section number, incorrect header comments, missing function prototypes if functions are used, missing signature line … etc