EECS 560 Lab 1 – Implementation of Singly Linked List

Prof.: Dr.Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

Maximum possible points: 100

Due date: 02/03/2020, 11:59 pm (Monday) - Tuesday lab
02/05/2020, 11:59 pm (Wednesday) - Thursday lab

Lab Instructions:

The purpose of this Lab is to review and practice implementation of a Singly Linked List and related operations in C++. This lab will help you brush up on your basic C++ concepts. Your program is expected to produce output as shown in the Expected Output section of this document. You are to read in the numbers from a data file of integers, one at a time, and put them into a Singly linked list. The list must be built as you read in the numbers. **You are not allowed to use a vector or array. Your program should handle the duplicate numbers in the following way. Duplicate values can be inserted in the list. When the delete function is called, it should remove the last occurrence of the duplicate number.**

Operations on Singly Linked List:

As you execute the code, your program should read the data from the input file (e.g., data.txt) and store the data in a singly linked list. After that your program should have a simple menu like as shown below.

Please choose one of the following commands:

- IsEmpty(): Returns true if the list is empty or the head node is NULL.
- Length(): Return the length of linked list.
- Insert(x): Inserts an element at the start of linked list.
- Delete(x): Deletes a given element. If a duplicate is present, deletes the last occurrence of the duplicate value.
- DeleteDuplicates(): Removes all the duplicate elements in the list. The output should be a list of unique elements. All duplicates except the first occurrence of each element are to be removed.
- Find(x): Searches for an element and returns True if found or False if not found.
- FindNext(x): Finds and prints the next element of a given number. The cases below need to be checked.
  1) Searches for the next element of x and prints it. Prints None if x is the last element.
  2) If there is a duplicate x value, prints the next element after the first occurrence of x.

3) Always perform find(x) before finding next. If x is found, then proceed with finding the next element.

- Print(): Prints the list.
- ReverseList(): Prints the reversed list.
- PrintAt(x): Prints the element at a given position x.
- Exit(): Exits the program.

All the Invalid cases must be handled for all of the menu options, as well as input file data (numeric). When you are finished, submit the final project along with a Makefile so that we may build an executable and run your code. Make sure to zip or tar.gz everything together.

Expected Output:

Sample data.txt: 1 2 23 34 23 2 4 5 2 67

Your program should produce the menu below and should work as per the sample given below. However, we will test your code with inputs other than the one shown below.

```
--------------------------------------------------------------------------------------------------------

Choose one operation from the options below:

    1. Is Empty
    2. Length
    3. Insert
    4. Delete
    5. Delete Duplicates
    6. Find
    7. Find Next
    8. Print
    9. ReverseList
    10. Print At
    11. Exit

> 8

List:  67 2 5 4 2 23 34 23 2 1

.........................................................

Choose one operation from the options below:

    1. Is Empty
    2.  Length
    3.  Insert
    4.  Delete
    5.  Delete Duplicates
    6.  Find
    7.  Find Next
    8.  Print
    9.  Reverse List
    10. Print At
    11. Exit

> 9

Reversed list:  1 2 23 34 23 2 4 5 2 67

.........................................................

Choose one operation from the options below:

    1.  Is Empty
    2.  Length
    3.  Insert
    4.  Delete
    5.  Delete Duplicates
    6.  Find
    7.  Find Next
    8.  Print
    9.  Reverse List
    10. Print At
    11. Exit

>2

The length of the list is:  10

.........................................................

Choose one operation from the options below:

    1.  Is Empty
    2.  Length
    3.  Insert
    4.  Delete
```

```
     5.  Delete Duplicates
     6.  Find
     7.  Find Next
     8.  Print
     9.  Reverse List
     10. Print At
     11. Exit

>10

Choose a position to print element:

>9

The element at the 9th position is:  2

........................................................

Choose one operation from the options below:

     1.  Is Empty
     2.  Length
     3.  Insert
     4.  Delete
     5.  Delete Duplicates
     6.  Find
     7.  Find Next
     8.  Print
     9.  Reverse List
     10. Print At
     11. Exit

>3

Choose a number to be inserted:

>11

> 11 is inserted.

.........................................................

Choose one operation from the options below:
     1.  Is Empty
     2.  Length
     3.  Insert
     4.  Delete
     5.  Delete Duplicates
     6.  Find
     7.  Find Next
     8.  Print
     9.  Reverse List
     10. Print At
     11. Exit

>8

>  11 67 2 5 4 2 23 34 23 2 1

.........................................................

Choose one operation from the options below:

     1.  Is Empty
     2.  Length
     3.  Insert
     4.  Delete
     5.  Delete Duplicates
     6.  Find
     7.  Find Next
     8.  Print
     9.  Reverse List
     10. Print At
     11. Exit

>4

Choose a number to be deleted from the list:

>23

The last occurrence of 23 has been deleted from the list

.........................................................

Choose one operation from the options below:
     1.  Is Empty
     2.  Length
     3.  Insert
     4.  Delete
     5.  Delete Duplicates
     6.  Find
     7.  Find Next
     8.  Print
     9.  Reverse List
     10. Print At
     11. Exit

>8

>  11 67 2 5 4 2 23 34 2 1

.........................................................

Choose one operation from the options below:

     1.  Is Empty
     2.  Length
     3.  Insert
     4.  Delete
     5.  Delete Duplicates
     6.  Find
     7.  Find Next
     8.  Print
     9.  Reverse List
     10. Print At
     11. Exit

>1

>  The list is not empty

.........................................................

Choose one operation from the options below:

     1.  Is Empty
     2.  Length
     3.  Insert
     4.  Delete
     5.  Delete Duplicates
     6.  Find
     7.  Find Next
     8.  Print
     9.  Reverse List
     10. Print At
     11. Exit

>6

Enter the number to find:

>56

> 56 is not found in the list.

.........................................................

Choose one operation from the options below:

     1.  Is Empty
     2.  Length
     3.  Insert
     4.  Delete
     5.  Delete Duplicates
     6.  Find
     7.  Find Next
     8.  Print
     9.  Reverse List
     10. Print At
     11. Exit

>7

Enter the number to find its next element:

>2

23 is next after 2.

.........................................................

Choose one operation from the options below:
     1.  Is Empty
     2.  Length
     3.  Insert
     4.  Delete
     5.  Delete Duplicates
     6.  Find
     7.  Find Next
     8.  Print
     9.  Reverse List
     10. Print At
     11. Exit

>7

Enter the number to find its next element:

>56

There is no element 56 in list.  Hence there is no next element.

.......................................................

Choose one operation from the options below:

     1.  Is Empty
     2.  Length
     3.  Insert
     4.  Delete
     5.  Delete Duplicates
     6.  Find
     7.  Find Next
     8.  Print
     9.  Reverse List
     10. Print At
     11. Exit

>5

>Duplicates deleted

.......................................................

Choose one operation from the options below:
     1.  Is Empty
     2.  Length
     3.  Insert
     4.  Delete
     5.  Delete Duplicates
     6.  Find
     7.  Find Next
     8.  Print
     9.  Reverse List
     10. Print At
     11. Exit

>8

11 67 2 5 4 23 34 1

.........................................................

Choose one operation from the options below:

     1.  Is Empty
     2.  Length
     3.  Insert
     4.  Delete
     5.  Delete Duplicates
     6.  Find
     7.  Find Next
     8.  Print
     9.  Reverse List
     10. Print At
     11. Exit

>11

>Program execution completed!
```

Execution Instructions:

GDB/DDD can be used to debug your code. Make sure to use Valgrind to execute your code to check for any memory leaks/segmentation faults.

Grading rubric:

- Full grade: Program should execute without any issues with all the options executed and with no memory leaks.
- Points will be taken for any execution errors, such as memory leaks, segmentation/program abort issues and missing handling of invalid cases.
- Programs that are compiled but do not execute will earn in the range of 0 to 50% of the possible points. Your grade will be determined based on the program design and the options implemented in the code.

Submission instructions:

- All files, i.e., the source files and Makefile, should be zipped in a folder.
- Include a ReadMe.txt if your code requires any special instructions to run.
- The naming convention of the folder should be LastName_Lab1.zip (or .tar or .rar or .gz).
- Email it to: chiranjeevi.pippalla@ku.edu or prashanthi.mallojula@ku.edu (your respective lab instructor) with subject line EECS 560 Lab1.
- Your program should compile and run on the **Linux machines** in **Eaton 1005D using g++.**