



Preparatory data Structure (CSCI 591)



Project - III

Implmenting and Excercising a Linked List

Submitted By: Taddese Erba

February 11, 2020

St. Cloud state university

Department of Computer Science

Taddese Erba
Section – I
Project – Three
Due: January 30, 2020

Design Document

Introduction

A linked list is a linear structure of ordered objects that are stored at random memory locations and linked together by pointers. An object of a linked list is called a node. Every node in a linked list contains two fields, the data field and the reference field. The data field stores the data whereas the reference field is a pointer that connects to the next node. Unlike arrays, linked lists can store an indefinite number of items that can be accessed by referring to its node.

This project will implement important operations in a linked list using the `LinkedList` class. The `LinkedList` class consists of the `List.h` header file where all the private and public variables and functions declared and the `List.cpp` implementation file. The `main.cpp` file is the testing file for all the implemented class functions.

Data Structure

This program has three distinct files. The `List.h` file contains all the declaration of the required functions and a few function decoration (implementation). It is the framework for the `LinkedList` class implementation. It consists of two private objects, the `struct Node` object which is used to hold the two main components of a node and the `Node * getNode(Item entry, Node * list)` function used to declare and initialize the node. In addition to the two private objects, the header file contains eleven public functions which include two constructors, one destructor, seven operational functions, and one friend function.

Taddese Erba
Section – I
Project – Three
Due: January 30, 2020

Functions

As mentioned in the Data Structure section, there are eleven functions in this project. The first two functions, the `LinkedList()` and `LinkedList(const LinkedList& source)` are constructors. The `LinkedList()` constructor function is used to initialize the class. The constructor `LinkedList(const LinkedList& source)` is a copy constructor that is used to copy the elements of the list. The third function, `~LinkedList()`, is a destructor. It is used to delete all the nodes, deallocate the memory, and return it to the operating system. The `void re_Initialize()` function is used to re-initialize the linked lists to empty. The `void insert(Item item)` function is used to insert items into the list. The `void remove(Item item)` is used to remove a node from the list. The `void operator = (LinkedList source)` is a function that overloads the assignment operator (`=`) to be used in the assignment operation involving copying the elements of the list. The `bool isEmpty()` function returns true if the list is empty or false otherwise. The `int listLength()` function will count each node in the list and return the size of the list (number of nodes). The `bool isPresent(Item target)` will check if an item is in the list and returns true. If the item is not found it returns false to `main()`. The `Item kthValue(int item)` function will return the k^{th} node of the list. If the node is not found, or the list is empty, it returns nothing. The `friend ostream& operator << (ostream& out_s, const LinkedList& l)` is a friend function that is used to overload the ostream operator (`<<`) for the purpose of printing all the elements of the list. For convenience reasons, the friend function is implemented in the header file where it was declared but outside of the class `LinkedList`.

Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

The Main Program

As a testing function, where the implementation is tested, there are many things going on in the main() function. To keep things simple, I will talk only about the main components of the main() function. The key frameworks in the main() function are the instantiation of the class LinkedList and representation of the key operations by a menu system. There are ten main menus from which the user can choose to perform an operation. The menus are represented by alphabets that are closely related to the operation followed by the name of the operation as in ***I -- Insert Item*** and ***R -- Remove Item***. The menus are continuously displayed after each operation until the user chooses to quit the program. A switch statement will track each choice of the user and perform the necessary operation accordingly. It may also worth mentioning the `bool searchArray(const char [], int, char)` function that is the part of the main() function that is used to search the array of constants that hold the alphabets designated to the menu. This enables that if the user enters a choice that is not available, the program can display the necessary message and exit the program.

Code listing

a. The header file (List.h)

```

1  /*
2      **** This is the "List.h" header file. ****
3      It contain the following three main parts.
4          1. The class LinkedList
5              => This class hosts:
6                  --> The private data type struct
7                  --> The private function get_node()
8                      to declare and initialize
9                      the struct object.
10                 --> The public constructors
11                     - LinkedList() declaration and implimentation
12                     - LinkedList(const LinkedList& source) declaration.
13                 --> The declaration of eight public functions & one friend function.
14          2. The implementation of the friend function outside the class.

```

Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

```

15     Precondition for this program to run properly:
16     => The program works for integer data only.
17     Postcondition for the program
18     => It can handle any number of items.
19     => It lists the items from smallest to the largest.
20     => It performs the following functions:
21         -> retaining the copy of the original data
22         -> inserting new item
23         -> deleting an item
24         -> re-initializing the list to empty
25         -> searching and returning the kth value of the list
26         -> looking up for an item in the list
27         -> checking if the list content for emptiness
28         -> print the list on the screen
29         -> counting the number of items in the list.
30  */
31  #include <iostream>
32  #ifndef _LIST
33  #define _LIST
34  #include <ostream>      //for the implementation of the friend function.
35  using namespace std;
36  class LinkedList{
37      typedef int Item;    //type defination decoration.
38      private:
39          struct Node{    // for the linked lists
40              Item item;

```

Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

```

41     Node * next;
42 };
43     Node * first;
44     Node * getNode(Item entry, Node * list);
45 public:
46     LinkedList(){           //default constructor
47         first = NULL;
48     }
49     LinkedList(const LinkedList& source);    // copy constructor
50     ~LinkedList();    // destructor
51     // to re-initialize the list to empty. The make_empty() function
52     // is absent because it will have the same function as re_initialize()
53     void re_Initialize();
54     void insert(Item item);           //to insert items to the list
55     void remove(Item item);           //to remove items from the list
56     void operator = (LinkedList source); // "=" Operator overloading
57     bool isEmpty();           //to check if the list is empty or not
58     int listLength();           //to get the number of nodes in the list
59     bool isPresent(Item target); //to check if an item is in the list
60     Item kthValue(int item);     //to access the kth item of the list
61     //the friend function is used for the purpose of
62     // "<<" operator overloading.
63     friend ostream& operator << (ostream& out_s, const LinkedList& l);
64 };
65 // Implimentation of the friend function.
66 ostream& operator << (ostream& out_s, const LinkedList& l){

```

```

67     LinkedList::Node * ptr;
68     ptr = l.first;
69     while(ptr!=NULL){
70         out_s << ptr -> item << ' ';
71         ptr = ptr -> next;
72     }
73     return out_s;
74 }
75 #endif //end of header file definition.
76 /* =====*/

```

Taddese Erba
Section – I
Project – Three
Due: January 30, 2020

b. The implementation file (List.cpp)

```
1  /*
2      **** This is the "List.cpp" implementation file ****
3      This file implements all the functions defined in the
4      "List.h" header file. Hence it is the file for class
5      implimentation.
6  */
7  #include <iostream>
8  #include "List.h"
9  #include <cstdint>
10
11 using namespace std;
12 ■ LinkedList::Node * LinkedList::getNode(Item entry, Node * list){
13     Node * temp;
14     temp = new Node;
15     temp->item = entry;
16     temp->next = list;
17     return temp;
18 }
19 ■ LinkedList::LinkedList(const LinkedList& source){
20     Node * ptr;
21     Node * last;
22     if(source.first == NULL)
23         first = NULL;
24 ■     else{
25         first = getNode(source.first -> item, NULL);
26         last = first;
```

Taddese Erba
Section – I
Project – Three
Due: January 30, 2020

```
27     ptr = source.first -> next;
28     while(ptr != NULL){
29         last->next = getNode(ptr->item, NULL);
30         last = last->next;
31         ptr = ptr->next;
32     }
33 }
34 }
35 LinkedList::~LinkedList(){
36     Node * temp;
37     while(first != NULL){
38         temp = first;
39         first = first->next;
40         delete temp;
41     }
42 }
43 void LinkedList::operator =(LinkedList source){
44     Node * ptr;
45     Node * last;
46     LinkedList empty;
47     if(&source != this){
48         empty.~LinkedList();
49
50         first = getNode(source.first -> item, NULL);
51         last = first;
52         ptr = source.first -> next;
```


Taddese Erba
Section – I
Project – Three
Due: January 30, 2020

```
53   while(ptr != NULL){  
54       last->next = getNode(ptr->item, NULL);  
55       last = last->next;  
56       ptr = ptr->next;  
57   }  
58   }  
59   else return;  
60 }  
61 int LinkedList::listLength(){  
62     Node * ptr;  
63     int count = 0;  
64     ptr = first;  
65     while(ptr != NULL){  
66         ++count;  
67         ptr = ptr->next;  
68     }  
69     return count;  
70 }  
71 bool LinkedList::isEmpty(){  
72     Node * ptr;  
73     ptr = first;  
74     if(ptr == NULL)  
75         return true;  
76     else return false;  
77 }  
78 bool LinkedList::isPresent(Item target){
```

Taddese Erba
Section – I
Project – Three
Due: January 30, 2020

```
79     Node * ptr;  
80     ptr = first;  
81     while(ptr != NULL && ptr -> item != target){  
82         ptr = ptr -> next;  
83     }  
84     return ptr != NULL;  
85 }  
86 void LinkedList::insert(Item newItem){  
87     Node* prev;  
88     if(first == NULL || newItem < first -> item)  
89         first = getNode(newItem, first);  
90     else{  
91         prev = first;  
92         while(prev -> next != NULL && prev -> next -> item < newItem)  
93             prev = prev -> next;  
94         prev -> next = getNode(newItem, prev -> next);  
95     }  
96 }  
97 void LinkedList::remove(Item oldItem){  
98     Node * prev;  
99     Node * temp;  
100    prev = first;  
101    if(first -> item == oldItem){  
102        first = first -> next;  
103        delete prev;  
104    }
```

Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

```

105 else{
106     while(prev -> next -> item < oldItem)
107         prev = prev -> next;
108     temp = prev -> next;
109     prev -> next = temp -> next;
110     delete temp;
111 }
112 }
113 LinkedList::Item LinkedList::kthValue(Item k){
114     Node * prev;
115     prev = first;
116     for(int i = 0; i < k; i++)
117         prev = prev -> next;
118     return prev -> item;
119 }
120 void LinkedList::re_Initialize(){
121     Node * prev;
122     prev = first;
123     if(prev == NULL){}
124     else{
125         LinkedList empty;
126         empty.~LinkedList();
127     }
128 }
129 /* =====*/

```

c. The testing file (main.cpp)

```

1  /*
2      **** This is the "main.cpp" testing file ****
3      This file tests the validity of the class implimentation
4      functions. The "main.cpp" has the following major duties:
5      1. It provides the user with a menu choice to enable them
6          to choose from the available menus. It repeats the menu
7          once the choosen task is completed and waits for the second
8          choce until the user chooses to quit the program.
9      2. It intializes the class LinkedList and calls its member functions
10         to perform the desired operations. Once the operation is over,
11         it anounces the result of that particular operation (choice).
12      3. It declare, implement, and run a function called
13         searchArray(const char [], int, char) that searches for the the
14         presence of the choice entered by a user in a constant array.
15         If the search is successful, the choice is performed.
16         If the search is unsuccessful, it displays a message accordngly.
17  */

```

Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

```

18 #include <iostream>
19 #include "List.h"
20 #include <iomanip>
21 #include "List.cpp"
22 #include <ostream>
23 using namespace std;
24 bool searchArray(const char [], int, char);
25 int main(int argc, const char * argv[]){
26     char ch;
27     const char array[] = {'c', 'e', 'i', 'm', 'l', 'n', 's', 'q', 'r', 'w'};
28     int item;
29     bool in, ck;
30     LinkedList source, list, list2;
31     cout << " This program will perform the following"
32     << "tasks.\n You must choose and enter the task"
33     << "\n you want to perform according to the \n instructions"
34     << "in the lists\n";
35     cout << " =====> << endl;
36     cout << " Please choose from the list below." << endl;
37     cout << " =====> << endl;
38     cout << " => I -- Insert Item"<<setw(30)<<"=> R -- Remove Item\n"
39     << " => E -- Check Emptiness"<<setw(25)<<"=> C -- Copy Items\n"
40     << " => L -- Lookup an Item"<<setw(27)<<"=> N -- Count Items\n"
41     << " => S -- Search Value"<<setw(32)<<"=> W -- Print Contents\n"
42     << " => M -- Make Empty"<<setw(31)<<"=> Q -- Exit Program" << endl;
43     cout << " =====> << endl;

```

Taddese Erba
Section – I
Project – Three
Due: January 30, 2020

```
44     cin >> ch;
45     in = searchArray(array, sizeof(array), ch);
46     if(in == false){
47         cout << " The choice you entered doesn't exist.\n";
48         cout << " See you later.\n Goodbye!"<< endl;
49         exit(0);
50     }
51     else{
52         while(ch){
53             switch(ch){
54                 case 'i':
55                     cout << "Enter an item to insert: ";
56                     cin >> item;
57                     list.insert(item);
58                     list2.insert(item);
59                     cout << "Item "<< item<<" inserted successfully."<<endl;
60                     break;
61                 case 'r':
62                     cout << "Enter an item to remove: ";
63                     cin >> item;
64                     list.remove(item);
65                     cout << "Item "<< item<<" removed successfully."<<endl;
66                     if(list.isEmpty() == true)
67                         list.~LinkedList();
68                     break;
```

Taddese Erba
Section – I
Project – Three
Due: January 30, 2020

```
69     case 'e':
70         ck = list.isEmpty();
71         if(ck == true)
72             cout << "List is empty."<<endl;
73         else
74             cout << "List is not empty."<<endl;
75         break;
76     case 'c':
77         cout << "Copy of the initial list:\n";
78         source = list2;
79         cout << source << endl;
80         break;
81     case 'l':
82         cout << "Enter an item to check: ";
83         cin >> item;
84         if(list.isPresent(item) == true)
85             cout << "Item "<<item <<" is in the list.\n";
86         else
87             cout << "Item "<<item <<" is not in the list.\n";
88         break;
89     case 'n':
90         cout << "The list has "<<list.listLength()<<" items"<<endl;
91         break;
```

Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

```

92         case 's':
93             cout << "Enter the index of the item you want to access: ";
94             cin >> item;
95             cout << "The element at index "<<item<<" is: ";
96             cout << list.kthValue(item);
97             break;
98         case 'w':
99             if(list.isEmpty())
100                 cout << "The list is empty\n";
101             else{
102                 cout << "Here are the items in the current list:\n";
103                 cout << list << endl;
104             }
105             break;
106         case 'm':
107             list.~LinkedList();
108             cout << "List is re-initialized to empty.\n";
109             break;
110         case 'q':
111             cout << "You chose to quit the program.\n";
112             cout << "See you later!";
113             exit(0);
114             break;
115     }

```

```

116         cout << " Please choose from the list below." << endl;
117         cout << " =====" << endl;
118         cout << " => I -- Insert Item"<<setw(30)<<"=> R -- Remove Item\n"
119             << " => E -- Check Emptiness"<<setw(25)<<"=> C -- Copy Items\n"
120             << " => L -- Lookup an Item"<<setw(27)<<"=> N -- Count Items\n"
121             << " => S -- Search Value"<<setw(32)<<"=> W -- Print Contents\n"
122             << " => M -- Make Empty"<<setw(31)<<"=> Q -- Exit Program" << endl;
123         cout << " =====" << endl;
124         cin >> ch;
125     }
126
127 }
128 return 0;
129 }

```

Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

```

130 bool searchArray(const char A[], int n, char ch){
131     int i = 0;
132     bool found = false;
133     while (i < n ){
134         if (ch == A[i] || tolower(ch) == A[i])
135             found = true;
136         i++;
137     }
138     return found;
139 }

```

Test Results

1. Tests result for insert() function.

```

This program will perform the following tasks.
You must choose and enter the task
you want to perform according to the
instructions in the lists
=====
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====
i 22
Enter an item to insert: Item 22 inserted successfully.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====
i 23
Enter an item to insert: Item 23 inserted successfully.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====

```


Taddese Erba
Section – I
Project – Three
Due: January 30, 2020

```
i 32
Enter an item to insert: Item 32 inserted successfully.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====

i 33
Enter an item to insert: Item 33 inserted successfully.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====

i 18
Enter an item to insert: Item 18 inserted successfully.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====
```

Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

```
i 17
Enter an item to insert: Item 17 inserted successfully.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====

i 15
Enter an item to insert: Item 15 inserted successfully.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====

i 40
Enter an item to insert: Item 40 inserted successfully.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====

i
Enter an item to insert: 45
Item 45 inserted successfully.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====
```

Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

```
W
Here are the items in the current list:
15 17 18 22 23 32 33 40 45
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====
```

2. Tests result for remove() function.

```
r 15
Enter an item to remove: Item 15 removed successfully.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====

r 23
Enter an item to remove: Item 23 removed successfully.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====

r 45
Enter an item to remove: Item 45 removed successfully.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====
```

Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

```
W
Here are the items in the current list:
17 18 22 32 33 40
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness       => C -- Copy Items
=> L -- Lookup an Item        => N -- Count Items
=> S -- Search Value          => W -- Print Contents
=> M -- Make Empty            => Q -- Exit Program
=====
```

3. Tests result for isEmpty() function.

```
e
List is not empty.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness       => C -- Copy Items
=> L -- Lookup an Item        => N -- Count Items
=> S -- Search Value          => W -- Print Contents
=> M -- Make Empty            => Q -- Exit Program
=====
W
Here are the items in the current list:
17 18 22 32 33 40
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness       => C -- Copy Items
=> L -- Lookup an Item        => N -- Count Items
=> S -- Search Value          => W -- Print Contents
=> M -- Make Empty            => Q -- Exit Program
=====
```

Taddese Erba
Section – I
Project – Three
Due: January 30, 2020

4. Tests result for copy () function.

```
C
Copy of the initial list:
15 17 18 22 23 32 33 40 45
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====

W
Here are the items in the current list:
17 18 22 32 33 40
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====
```

Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

5. Tests result for isPresent() function.

```

1
Enter an item to check: 22
Item 22 is in the list.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====
1
Enter an item to check: 17
Item 17 is in the list.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====
1 34
Enter an item to check: Item 34 is not in the list.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====

```

Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

```

w
Here are the items in the current list:
17 18 22 32 33 40
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness       => C -- Copy Items
=> L -- Lookup an Item        => N -- Count Items
=> S -- Search Value          => W -- Print Contents
=> M -- Make Empty            => Q -- Exit Program
=====
  
```

6. Tests result for listLength() function.

```

n
The list has 6 items
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness       => C -- Copy Items
=> L -- Lookup an Item        => N -- Count Items
=> S -- Search Value          => W -- Print Contents
=> M -- Make Empty            => Q -- Exit Program
=====
w
Here are the items in the current list:
17 18 22 32 33 40
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness       => C -- Copy Items
=> L -- Lookup an Item        => N -- Count Items
=> S -- Search Value          => W -- Print Contents
=> M -- Make Empty            => Q -- Exit Program
=====
i
Enter an item to insert: 35
Item 35 inserted successfully.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness       => C -- Copy Items
=> L -- Lookup an Item        => N -- Count Items
=> S -- Search Value          => W -- Print Contents
=> M -- Make Empty            => Q -- Exit Program
=====
  
```

Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

```
i
Enter an item to insert: 15
Item 15 inserted successfully.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====

w
Here are the items in the current list:
15 17 18 22 32 33 35 40
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====

n
The list has 8 items
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====
```


Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

7. Tests result for kthValue() function.

```

S
Enter the index of the item you want to access: 4
The element at index 4 is: 32 Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====
W
Here are the items in the current list:
15 17 18 22 32 33 35 40
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====
S
Enter the index of the item you want to access: 6
The element at index 6 is: 35 Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====
S
Enter the index of the item you want to access: 9
The element at index 9 is:
-----
Process exited after 1626 seconds with return value 3221225477
Press any key to continue . . .

```

Note: If we try to access an index that is not in the list the the function will exhaust searching that index and exit the program as shown in the above screenshoot.

Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

8. Tests result for write () function

I am not running a separate test for the write () function since I am running it for almost every other step as part of checking the program to see if it is doing what it supposed to do. Please look at the end of the other programs where I occasionally run the write() function.

9. Tests result for makeEmpty () function

```

w
Here are the items in the current list:
12 15 26 35 44
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====
m
List is re-initialized to empty.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====
w
The list is empty
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====

```

Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

10. Tests result for quit

```
q
You chose to quit the program.
See you later!
-----
Process exited after 134.7 seconds with return value 0
Press any key to continue . . .
```

Moreover, if the user enters a choice that is not listed the program will announce that and exit as shown below.

```
This program will perform the following tasks.
You must choose and enter the task
you want to perform according to the
instructions in the lists
=====
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness       => C -- Copy Items
=> L -- Lookup an Item        => N -- Count Items
=> S -- Search Value          => W -- Print Contents
=> M -- Make Empty            => Q -- Exit Program
=====
t
The choice you entered doesn't exist.
See you later.
Goodbye!

-----
Process exited after 3.24 seconds with return value 0
Press any key to continue . . .
```

Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

User document

This program can perform different tasks on a linked list as shown in the menu below. In order to run the program, you must perform the following steps.

- ☞ The program name is `main.cpp`. on the terminal enter the following command to compile and run the program.
`g++ -o main main.cpp`
- ☞ The program will compile and open the following window:

```
This program will perform the following tasks.
You must choose and enter the task
you want to perform according to the
instructions in the lists
=====
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====
```

- ☞ Once the window opens, make a choice from the displayed menu. For example to insert an item type `i` or `I` and then enter.

```
i
Enter an item to insert:
```

- ☞ Next, type the item you want to insert and then enter. For example, type `15` and enter.

```
i
Enter an item to insert: 15
```

- ☞ The program will announce that the item is entered successfully and display the menu to make the next choice.

Taddese Erba
 Section – I
 Project – Three
 Due: January 30, 2020

```

Item 15 inserted successfully.
Please choose from the list below.
=====
=> I -- Insert Item           => R -- Remove Item
=> E -- Check Emptiness      => C -- Copy Items
=> L -- Lookup an Item       => N -- Count Items
=> S -- Search Value         => W -- Print Contents
=> M -- Make Empty           => Q -- Exit Program
=====
  
```

- ☞ If you want to repeat the insert repeat the above procedure; otherwise make the next selection.
- ☞ The program will perform in the same manner for all other tasks as for insert. Hence, all the other eight functions will perform in the same manner.
- ☞ Feel free to play around with the other choices (alphabets) and see what the program is meant to do.
- ☞ If you wish to exit the program, type q (Q) and enter.

```

q
You chose to quit the program.
See you later!
-----
Process exited after 794.8 seconds with return value 0
Press any key to continue . . .
  
```

- ☞ Now you can close the window.

Here are very important points while using this program

1. You must insert integer values only. If you try to enter something else other than an integer, the program may crash.
2. Do not try to access the ends of the list. If your list has 3 nodes only and you try to access the 4th node, the program will stop and exit.
3. The program will save a copy of the current list you are working with. You can just type c(C) and access that copy. Of course, once you exit the program that copy will not exist.

Taddese Erba
Section – I
Project – Three
Due: January 30, 2020

Summery

The project implements a linked list operation such as inserting a new item, removing an item from the list, making the list empty, checking for the presence of an item, displaying a copy of the original item that contains all the elements of the list and so on. The knowledge of linked lists and their implementation is very crucial because data are often stored and retrieved as lists.

This program can further be improved by making the necessary changes to make the program accommodate various types of data such as characters and strings. This will make the program a more useful data structure where we can store important records such as student records. Furthermore, data could also be made available as a file and a permanent copy of that file is kept with all current updates included while we still have the old data for reference.

I have gained a significant level of confidence and the necessary knowledge to work with linked lists by completing this project. I believe, this project is one of the projects I would keep working on and refer to the most even the future programming computations.