



CSE 30: Data Structures Laboratory 4

Due Date: Oct 4, 2020 at 11:59 pm

Introduction

In this lab we will be working with linked lists. There is no separate lab project in GitLab/CodeSync so you should just add the required functionality to your Lab 3 workspace.

Tasks

Setup

Create a file called `LinkedList.h` in your `inc` folder. The following is the support code provided, which should go into the `LinkedList.h` file.

```
struct LinkedList {
    Node* head;

    LinkedList(){
        head = NULL;
    }

    void append(int value){

        if (head == NULL){
            head = new Node(value);
        }
        else{
            // Allocate a new node with value
            // Go to the last element of the list
            // Make the next of the last element point to the new node

            Node* newNode = new Node(value);

            Node* temp = head;
            while(temp->next != NULL){
                temp = temp->next;
            }
            // At this point, temp is pointing to the last existing element of list
```

```

        temp->next = newNode;
    }
}

void insert(int index, int value) {
    // Provide your code here
}

int get(int index){
    // Provide your code here
}

void set(int index, int value){
    // Provide your code here
}

void print (){
    Node* temp = head;

    while (temp != NULL) {
        cout << temp->data << endl;
        temp = temp->next;
    }
}

~LinkedList(){
    Node* temp = head;

    while(temp != NULL){
        temp = temp->next;
        delete head;
        head = temp;
    }
}
};

```

Add the following lines to your `app.cpp` file:

```

LinkedList myList;

for (int i = 0; i < 6; i++) {
    myList.append(i);
}

myList.insert(2, 77);

myList.insert(10, 89);

```

```
myList.append(101);

myList.set(0, 11);

cout << myList.get(2) << endl << endl;

myList.print();
```

Your tasks

The first function you are being asked to implement is `int get(int index)`. This function simply returns the value that appears in the linked list position specified by `index`.

The second function is `void set(int index, int value)`. Its job is to store `value` in the linked list position corresponding to `index`.

The last function to implement is `void insert(int index, int value)`. As the name suggests, it needs to insert the `value` at the `index`. It should not overwrite anything. If there is already something stored at `index`, it should be shifted to the right. If `index` is larger than the current size of the list, then it needs to be resized in order to accomodate. If there is a gap between the old size of the list, and the newly inserted `value`, that gap should be filled with 0s.

Your `void insert(int index, int value)` function should be thoroughly unit tested.