Lab #7

CS-2050

October 21, 2022

1 Requirements

In this lab, you will cover creating and managing a linked list with a placeholder (or "dummy") node at the head of the list. Test code is provided that you may optionally use. Please remember that, unless otherwise specified, all data structures in this course are 0-indexed. You are given the following struct definition in your starter code:

```
struct Node {
        void *data;
        Node *next;
};
```

1.1 makeList

```
Node * makeList();
```

Info: This function will return a pointer to an empty list, or NULL on failure.

1.2 getSize

```
int getSize(Node *head);
```

Info: This function will take a pointer to a linked list, and return the number of elements on the list. The size of the list does not include the placeholder node.

1.3 insertAtIndex

```
int insertAtIndex(Node *head, void *data, int index);
```

Info: This function will take a pointer to a linked list, a data element to insert on the list, and an index to insert the element at. You may assume no NULL data will be inserted onto the list. If the insertion was successful, the function **will return 0**, otherwise it **will return 1**. If the provided index is the size of the list, the element will be inserted at the end of the list.

1.4 removeFromHead

```
void * removeFromHead(Node *head);
```

Info: This function will take a pointer to a linked list, and remove and return the element at the head of the list. If the list is empty, it will return NULL.

1.5 getAtIndex

```
void * getAtIndex(Node *head, int index);
```

Info: This function will take a pointer to a linked list, and return the element at the given index **without removing it**. If the index is invalid, it will return NULL.

1.6 freeList

```
void freeList(Node *head);
```

Info: This function will take a pointer to a linked list, and free all memory allocated to the list, including the placeholder node at the head of the list.

Notice

Grading: 25 points

1. Write required *makeList* function
   * 3 points
2. Write required *getSize* function
   * 5 points
3. Write required *insert* function
   * 8 points
4. Write required *remove* function
   * 3 points
5. Write required *get* function
   * 3 points
6. Write required *free* function
   * 3 points