Lab #5

CS-2050

February 24, 2023

1 Requirements

In this lab, you will cover creating and maintaining linked lists. Your list implementation should **not** make use of any placeholder (or "dummy") nodes. The list pointer should always point to the **head** of the list. In this lab you are given the following struct definition:

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

1.1 makeList

```
int makeList(Node **list)
```

Info: This function will initialize the provided pointer to Node * to an empty list. It will always return 0.

1.2 getSize

```
int getSize(Node *list)
```

Info: This function takes a list, and returns the number of elements on the list. Note that if the list is empty, the number of elements on the list is 0.

1.3 insertAtTail

```
int insertAtTail(Node **list, void *data)
```

Info: This function takes a pointer to Node *, and inserts the given data at the tail of the list. It returns 0 if insertion was successful, or 1 if insertion failed.

1.4 removeFromHead

```
void * removeFromHead(Node **list)
```

Info: This function takes a pointer to Node *, and removes the element at the head of the list (if any). It will return the data which was removed from the list, or NULL if the list was empty. You may assume that no data on the list will be NULL.

1.5 freeList

```
void freeList(Node **list)
```

Info: This function takes a pointer to Node *, and frees the memory allocated to the list. After freeing, it sets the pointer to NULL. **Note that the data on the list is not considered part of the memory allocated to the list.**

Grading: 18 points

1. Write required *makeList* function
   * 2 points
2. Write required *getSize* function
   * 3 points
3. Write required *insertAtTail* function
   * 5 points
4. Write required *removeFromHead* function
   * 5 points
5. Write required *freeList* function
   * 3 points

Notice:

1. All of your lab submissions **must** include documentation to receive full points.
2. All of your lab submissions must compile under GCC using the $-Wall$ and $-Werror$ flags to be considered for a grade.
3. You are expected to provide proper documentation in every lab submission, in the form of code comments. For an example of proper lab documentation and a clear description of our expectations, see the lab policy document.