Lab #7

CS-2050 - Section B

Week of March 15, 2021

1 Requirements

This lab is intended to test your ability to work with abstract data types and interface functions. You will not be provided with a main file in your starter code, and any testing code you produce will not be graded. In this lab, you will produce a set of *interface functions* for a list type which employs the use of a **linked list data structure**.

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

typedef struct {
        Node *head;
        int size;
} List;
```

1.1 initList

```
List* initList();
```

Info: This function initializes and returns a *linked list*.

1.2 getSize

```
int getSize(List *list);
```

Info: This function takes a *linked list* and returns the number of elements on the list.

1.3 freeList

```
void freeList(List *list);
```

Info: This function takes a *linked list* and frees all memory allocated for the list. Remember that you *should not* free the user's data, as that *does not* belong to your library.

1.4 getAtIndex

```
void* getAtIndex(List *list, int index);
```

Info: This function takes a *linked list* and returns the object at the given index, or NULL on error.

1.5 insertAfter

```
int insertAfter(List *list, void *object, void *sentinel);
```

Info: This function takes a *linked list* and attempts to insert the given object *after* the specified sentinel object in the list. **If the sentinel object does not exist on the list, the object should be inserted at the end of the list**. It should return 1 on success and 0 on failure.

1.6 listContains

```
int listContains(List *list, void *object);
```

Info: This function takes a *linked list* and returns 1 if the given object is on the list, or 0 otherwise.

1.7 removeAtIndex

```
void* removeAtIndex(List *list, int index);
```

Info: This function takes a *linked list* and removes the object at the given index of the list. This function must return the object to the user.

2 Notice

Grading: Total 37 points

1. Write required *init* function
   * 4 points
2. Write required *get size* function
   * 1 point
3. Write required *free list* function
   * 8 points
4. Write required *get at index* function
   * 5 points
5. Write required *insert* function
   * 7 points
6. Write required *remove* function
   * 7 points
7. Write required *listContains* function
   * 5 points

Notice:

1. All of your lab submissions must compile under GCC using the $-Wall$ and $-Werror$ flags to be considered for a grade.
2. 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.