Lab #6

CS-2050 - Section D

Week of March 8, 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 starts at **a user defined index**.

```
typedef struct {
        void **array;
        int size;
        int maxSize;
        int startIndex;
} OffsetList;
```

1.1 initOffsetList

```
OffsetList* initOffsetList(int maxSize, int startIndex);
```

Info: This function initializes and returns *an OffsetList* with the specified maxSize and startIndex.

1.2 getOffsetListSize

```
int getOffsetListSize(OffsetList *list);
```

Info: This function takes *an OffsetList* and returns the number of elements on the list.

1.3 freeOffsetList

```
void freeOffsetList(OffsetList *list);
```

Info: This function takes *an OffsetList* and frees all memory allocated by the init function.

1.4 getOAtIndex

```
void* getOAtIndex(OffsetList *list, int index);
```

Info: This function takes *an OffsetList* and returns the object at the given index, or NULL on error.

1.5 insertInOffsetList

```
int insertInOffsetList(OffsetList *list, void *object, int index);
```

Info: This function takes *an OffsetList* and attempts to insert the given object at the given index in the list. It should return 1 on success and 0 on failure.

1.6 hasNoObjects

```
int hasNoObjects(OffsetList *list);
```

Info: This function takes *an OffsetList* and returns 1 if the list contains anything, or 0 if it does not.

1.7 lastIndexOf

```
int lastIndexOf(OffsetList *list, void *object);
```

Info: This function takes *an OffsetList* and returns the index of the last occurrence of the given object in the list, or -1 if it does not exist on the list.

1.8 removeFromList

```
void* removeFromList(OffsetList *list, int index);
```

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

2 Notice

Grading: Total 25 points

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