

# Lab #8

CS-2050

October 28, 2022

## 1 Requirements

In this lab, you will cover implementing and managing a queue ADT. In this lab, every function has an associated complexity in the form of Big-O notation. **Each function must perform at the complexity specified, and must not perform faster or slower than required.** The underlying structure of the queue is not defined, and you must derive the implementation from the complexity requirements listed above each function. For any details of the implementation not specified in the requirements, you are free to make your own design decisions. You are given the following struct definitions in your starter code:

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

struct Queue {
    Node *list;
    int size;
};
```

### 1.1 makeQueue

```
// O(1)
Queue * makeQueue();
```



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

### 1.2 getSize

```
// O(1)
int getSize(Queue *q);
```



**Info:** This function will take a pointer to a Queue, and return the number of elements on the Queue.

### 1.3 enqueue

```
// O(n)
int enqueue(Queue *q, void *data);
```



**Info:** This function will take a pointer to a Queue, as well as a data pointer. It will enqueue the data pointer onto the queue. It will return 0 on success, or 1 on failure.

## 1.4 deQueue

```
// O(1)
void * deQueue(Queue *q);
```



**Info:** This function will take a pointer to a Queue, and dequeue and return the next item on the queue, or NULL if the queue is empty. Remember that the user has no information about the implementation details of your ADT.

## 1.5 peek

```
// O(1)
void * peek(Queue *q);
```



**Info:** This function will take a pointer to a Queue, and return the next item on the queue without dequeuing it, or NULL if the queue is empty.

## 1.6 freeQueue

```
// O(n)
void freeQueue(Queue *q);
```



**Info:** This function will take a pointer to a Queue, and free all memory allocated to the Queue. Remember that the user's data does not belong to your ADT.

## Notice



### Grading: 20 points

1. Write required *makeQueue* function
  - \* 3 points
2. Write required *getSize* function
  - \* 1 point
3. Write required *enqueue* function
  - \* 5 points
4. Write required *deQueue* function
  - \* 5 points
5. Write required *peek* function
  - \* 3 points
6. Write required *free* function
  - \* 3 points