Lab 10 - section F

We'll assume the following structures:

```
typedef struct nodestruct {
   int item;
   struct nodestruct *next;
} Node;
typedef struct {
   int size, sum;
   Node *rear;
} Queue;
```

In this lab, you need to implement five functions in a circular-linked-list based on the given structure definitions above.

Note: no dummy nodes or doubly circular linked list should be used for this lab assignment.

Queue* initQueue(); // Return an empty queue, *O(1) complexity*

int enqueue(int, Queue*); // Add a new node with an integer whose value is between 1 and 10 into the queue and return an error code, *O(1) complexity*

int dequeue(Queue*, int*); // Remove a node from the queue, return the key value in the removed node by the integer pointer, and return an error code, *O(1) complexity*

float getAverage(Queue*); // Return the average value of all integers in the queue as a floating-point number, *O(1) complexity, note: there's a member "sum" in Queue structure*

void freeQueue(Queue*); // Free all space, *O(n) complexity*

You could create the main program in whatever way you like to test your functions. TAs will grade all your functions using a uniform testing program.

The assignment you need to submit should include everything but your main program, for example, your submission should look like this:

includes
Declarations of structures
Function prototypes
Function implementations

Grading Criteria:

initQueue function: 5 points

enqueue function: 15 points

dequeue function: 15 points

getAverage function: 5 points

freeQueue function: 5 points

All or nothing rule when grading: starting from lab 6, no partial credits will be given for incomplete/incorrect functions.

General note:

1. Command to compile your code in cmd window: gcc labx.c -Wall -Werror
2. If your code does not compile with “-Wall -Werror” flag, you will receive an automatic 0 for this assignment.
3. Changing the given function prototype or struct definition will lead to an automatic zero grade.
4. Using any global variables will lead to an automatic zero grade.
5. The implementation of the function should include comments describing what it is intended to do and how this function should be called. Example can be found in CS 2050 lab policy.
6. If your submission does not include a source file, you will receive an automatic zero grade.