3) (10 pts) DSN (Queues)

A regular queue only supports adding an item to the back/tail of the queue and removing the item at the front/head of the queue. A common method of implementing a queue is as a linked list with pointers to both the front/head and back/tail of the list. If this method is used, it's relatively easy to add the functionality of adding an item to the front/head of the queue and removing an item from the back/tail of the queue. Given below is a partial implementation of this data structure (commonly called a deque). Fill in the function that adds an item to the front/head of the queue. You may call the makeNode function, and you may assume that myDeque points to a deque that exists (though it may or may not be empty.)

```
#include <stdio.h>
#include <stdlib.h>
typedef struct node {
    int data;
    struct node* next;
} node;
typedef struct deque {
    node* front;
    node* back;
} deque;

deque* makeEmptyDeque() {
    deque* tmp = malloc(sizeof(deque));
    tmp->front = tmp->back = NULL;
    return tmp;
}

node* makeNode(int val) {
    node* tmp = malloc(sizeof(node));
    tmp->data = val;
    tmp->next = NULL;
    return tmp;
}

void addFront(deque* myDeque, int val) {



}
```

Computer Science Foundation Exam

May 20, 2023

Section B

ADVANCED DATA STRUCTURES

NO books, notes, or calculators may be used,
and you must work entirely on your own.

Name: ____

UCFID: ____

Question #	Max Pts	Category	Score
1	5	ALG	
2	10	ALG	
3	10	ALG	
TOTAL	25		

You must do all 3 problems in this section of the exam.

Problems will be graded based on the completeness of the solution steps and <u>not</u> graded based on the answer alone. Credit cannot be given unless all work is shown and is readable. Be complete, yet concise, and above all <u>be neat</u>. For each coding question, assume that all of the necessary includes (stdlib, stdio, math, string) for that particular question have been made.