2413 Assignment 3

Due Date
2-16-16 before lab

For this assignment you will be implementing a QUEUE ADT with a Doubly Linked List under the hood. You should continue the pattern for assignments by having a "queue.h" which contains the interface for the queue. "queue.c" contains the implementation and "main.c" is responsible for processing commands through standard in and calling the appropriate queue functions.

QUEUE ADT

1. queue type: (doubly linked list, needing a front pointer and back pointer to the list of nodes. (Each node has an element e, next pointer, and prev pointer))
2. createQueue: $void \rightarrow queue\_ptr$
3. destroyQueue: $queue\_ptr \rightarrow void$
4. enqueue: $queue\_ptr \times element \rightarrow void$
5. dequeue: $queue\_ptr \rightarrow element$
6. peek: $queue\_ptr \rightarrow element$

Your queue must be implemented as a doubly linked list. You can either start writing the queue from scratch or you can take your implementation of the stack and modify it to create a doubly linked list.

In order to test your queue implementation, we specify an interface through standard IO (scanf and printf). The user types in commands and the program executes them to produce the desired output.

1 Program IO

Input Commands on Standard IO

1. enqueue #1 #2

- Assuming #1 < #2, use a loop to push integers #1 through #2 inclusive onto the back of the queue in ascending order.

2. dequeue #1
   - perform the dequeue operation #1 number of times.
3. peek
   - print the integer at the front of the queue.
4. display
   - Displays the content of the queue from back to front.

1.1 Format For Output Per Input

enqueue #1 #2

```
Enqueued #1
Enqueued #1+1
Enqueued #1+2
...
Enqueued #2-1
Enqueued #2
```

dequeue #1

```
Dequeued #
....      (#1 number of dequeus performed)
Dequeued #
```

When the queue becomes empty then "Queue Is Empty" is printed on its own line.

Sample Commands:

```
enqueue 0 3
dequeue 6
```

Expected Output:

```
Enqueued 0
Enqueued 1
Enqueued 2
Enqueued 3
Dequeued 0
Dequeued 1
Dequeued 2
Dequeued 3
Queue Is Empty
Queue Is Empty
```

Sample Commands:

```
enqueue 0 3
peek
display
```

Expected Output:

```
Enqueued 0
Enqueued 1
Enqueued 2
Enqueued 3
Peeked 0
Display In Reverse
3
2
1
0
```

2 Using The Grading Script

Included in this assignment is the grading script I'm using to grade your assignments. The output I expect from your program for each test case is the reference folder. Your output needs to match the expected output exactly otherwise the grading script will not give you credit.

Instructions For Running The Grading Script

1. Compile: gcc *.c -o grade.exe
2. Change "roster.txt" to contain your last name, followed by a space, followed by your first name.

3. Create a directory named from your lastname followed by the first initial.
4. Copy your program into the directory.
5. run the grade script "grade grade" on windows or "./grade.exe grade" on *nix.