instructions.md

A3

Due: See dropbox details
Submit: .zip

Details

For this assignment you will be developing a C++ program which implements a priority queue data. Your program should implement a C++ class which will be used to represent your priority queue. You have been provided with the following files:

- `pqueue.h` Defines the interface your class should implement
- `main.cpp` This is a minimum file which contains the driving logic for your program
- `makefile` This is the makefile which can be used to build your program
- `pqueue.cpp` This is the implementation file for your queue. This file is currently empty
- `cmd.txt` This is the file that your program will read input from and determines what your program will do
- `output.txt` Contains the expected output for a correctly implemented program which runs the provided `cmd.txt`

Your Task

You are to look at the class defined in `pqueue.h` and create the corresponding implementation in `pqueue.cpp`. You may not change any of the existing declarations but you may add more functions as you deem necesarry. The priority of the node will be determined by its value. The lower the value of the node the higher the priority in the queue.

cmd File Format

The cmd file for this assignment consists of two columns of numbers. The first column indications the function that should be called, and the second column indications the operand that that function may take. If the function takes no operand then the entry will be blank. The first column number to function mapping is as follows

```
1    enq
2    deq
3    front
4    isEmpty
5    printq
```

How I will test your program

To test your program I will use your `pqueue.h` and `pqueue.cpp` files and combine them with my own `main.cpp` file. My version of `main.cpp` will be similiar to the one that is provided with a few extra grading features and test cases added in.

Deliverables

Add `pqueue.h`, `pqueue.cpp`, `main.cpp`, `makefile`, and `cmd.txt` to a zip file called `a3.zip` and submit that file to the dropbox. I will only accept submissions that meet these requirements.

Expections

- No use of any prebuilt queue libraries
- Your code should be well formatted, points may be taken for sloppy code
- Your output should match the output provided in output.txt
    - For this assignment, the results of printq() may differ from my output as it is possible to implement the priority queue in different ways
- You must submit a zip file called `a3.zip`
- Your code should be fairly robust and be able to handle obvious edge cases such as if I remove every element from the queue and begin to add new elements. You are responsible for comming up with and testing for possible edge cases.