

CSC331: Data Structures – BMCC Spring 2022
Professor Byron
Program #4 specifications: Priority queue
Due: 1159pm Monday 5-2-2022
No credit for late submission

Your task for this assignment is to develop a print scheduling application. The application will use an array-based heap to implement a priority queue data structure in C++. This will be accomplished utilizing a user-defined Heap class.

1. Implement a transaction-based priority queue data structure using C++. The program will be interactive. A prompt will be displayed requesting an input transaction. Each input transaction will be entered at the command line and results will be displayed on the console.
2. Each input transaction will represent either (1) a new print request transaction or (2) a "service" transaction. A new print request transaction will be represented by a string consisting of a priority code (1, 2 or 3), a sequential job number and a user name all separated by hyphens. Each print request is scheduled based on priority and job number. The priority was already determined by document size and user name. Priority 1 is the highest and priority 3 is the lowest. Each print request is placed in a priority queue and will be serviced by the next available printer.

A "service" transaction results in a print request being removed from the priority queue and the customer name being displayed on the console. A transaction containing the string "end-of-file" will signal the end of the input. Your program will display the number of print requests remaining in the queue when the "end-of-file" transaction is processed.

Sample input transactions and output expected using a priority queue:

```
2-01-guo
1-02-hannie
3-03-ian
service (output: 1-02-hannie)
service (output: 2-01-guo)
1-04-danny
3-05-eli
1-06-fan
service (output: 1-04-danny)
service (output: 1-06-fan)
end-of-file (output: There are 2 remaining print requests in the queue.)
```

3. The program will be run at the command prompt by navigating to the directory containing the executable version of the program after the program is compiled. Assume that each transaction is prepared properly. Use an array with a capacity for 12 print requests. Display “Error: queue is empty” and “Error: queue is full” messages as needed.
4. Your C++ program file should be named `csc331_prog4_lastname.cpp`. Your program should contain comments starting on line 1 of the program containing the following information:
 - a. course ID and section
 - b. your full name
 - c. the program file name
 - d. the program assignment number and due date
 - e. the program purpose

You are encouraged to add additional comments throughout the program that you feel might be helpful to the reader of your source code. Your program code indentation must be neat for readability.

5. Submit your C++ program file as an attachment to an email message to `kbyron@bmcc.cuny.edu` using a subject in this form: “`csc331_prog4_lastname`”. No other files are to be submitted.
6. Grading note: you will receive 2/3 credit if your program functions as a standard first-in-first-out (FIFO) queue using user-defined queue class or queue Standard Template Library (STL). For 4/3 credit, implement as described using a user-defined priority queue class with a small array implementation of a heap.

Sample input transactions and output expected using a FIFO queue:

```
2-01-guo
1-02-hannie
3-03-ian
service (output: 2-01-guo)
service (output: 1-02-hannie)
1-04-danny
3-05-eli
1-06-fan
service (output: 3-03-ian)
service (output: 1-04-danny)
end-of-file (output: There are 2 remaining print requests in the queue.)
```