Your task for this assignment is to develop a lawn mower repair 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 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 mower repair order transaction or (2) a "service" transaction. A new mower repair order transaction will be represented in three parts separated by hyphens: customer name (variable length), mower model year (4 digits) and a warranty code (y or n). Each new repair order is issued a sequential 2-digit ticket number (00 through 99). Each repair order will be scheduled based on priority and ticket. A warranty repair order (i.e., warranty code = 'y') is assigned priority 1 (highest). A non-warranty repair for a mower model less than 6 years old is assigned priority 2. A non-warranty repair order for a mower model older than 5 years is assigned a priority 3 (lowest). Each repair order is placed in a priority queue and will be serviced by the next available repair technician.

   A "service" transaction results in a mower repair order 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 mower repair orders remaining in the queue when the "end-of-file" transaction is processed.

Sample input transactions and output expected using a priority queue:

```
guo-2018-y
hannie-2014-n
ian-2004-y
service  (output: guo)
service  (output: ian)
danny-2012-n
eli-2017-n
fan-1992-y
service  (output: fan)
service  (output: eli)
ann-2013-y
bow-2016-n
cho-2013-n
service  (output: ann)
service  (output: bow)
service  (output: hannie)
end-of-file  (output: There are 2 remaining repair orders 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 10 mower repair orders. Display "Error: queue is empty" and "Error: queue is full" messages as needed.

4. Your C++ program file should be named csc331_section_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_section_prog4_lastname".

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:

```
guo-2018-y
hannie-2014-n
ian-2004-y
service  (output: guo)
service  (output: hannie)
danny-2012-n
eli-2017-n
fan-1992-y
service  (output: ian)
service  (output: danny)
ann-2013-y
bow-2016-n
cho-2013-n
service  (output: eli)
service  (output: fan)
service  (output: ann)
end-of-file  (output: There are 2 remaining repair orders in the queue.)
```