Assignment 4 – Priority Queue

Description:

Implement a priority queue capable of determining the order in which a group of people should be admitted to a concert based off of the number designated on their ticket. The highest priority should be given to those individuals with the lowest ticket numbers. Your program should read a file containing ticket information and print the order in which individuals should be admitted into the concert.

Your priority queue should be written using a C++ class, named PQueue, and the queue itself should be implemented as a linked list similar to how we implemented queues/stacks in lecture.

The following classes should be defined by your program:

`class Person` : Used to store information about each person (ticket # and name)
`class PQueue` : Priority queue used to serve people

The following public functions need to be implemented:

`Person::Person( int, string )`: Initialize the person
`PQueue::PQueue( void )` : Initialize the priority queue
`bool PQueue::empty( void )` : Test whether queue is empty
`int PQueue::size( void )`: Return size
`Person* PQueue::front( void )`: Access node at front of queue
`Person* PQueue::back( void )`: Access node at end of queue
`void PQueue::enqueue( Person* )` : Insert node into proper location in queue
`void PQueue::dequeue( void )` : Remove node from front of queue

** Do not change names, as it will interfere with grading.

Like your other programs, it will need to read information from a file. The file will be structured as follows:

```
int string
```

It will contain n lines with each line consisting of an integer representing the ticket number followed by one space followed by a string representing the ticketholder's name. The ticket number should be used to designate the priority assigned to the ticketholder.

Example:

```
$ cat input.txt
1 Billy
4 Mary
2 Jack
3 Shawn
$ ./a.out
Admitting Billy..
Admitting Jack..
Admitting Shawn..
Admitting Mary..
```

Due:

November 2nd, 2016 11:59 PM