CS 3024 Home Assignment 3 (20 points)

Sakai folder “Hash Tables, Heap, Priority Queue, Graphs” contains file **priority_queue.cpp** with the definition of class **Priority_queue** with **int** keys and using a max-heap.

Your task is to add to the class **Priority_queue** new public member functions:

void clear() resets the priority queue to empty by setting the current size to 0;

void decrease_key(int position, int new_key) changes the key value for the queue element at ***position*** and restores the max-heap property of the heap, please notice that this function should check whether the ***position*** is within range of the queue’s array, this feature is missing in the **increase_key()** function in the **priority_queue.cpp** file ☹, and it may be a good idea to add it to it as well;

void print_visual() prints the heap in the form of layered tree following the layers stored in the heap array.

Examples.
Empty queue will be printed as **Empty**;

Queue with a single element nnn will be rendered as **nnn**;

Queue stored as an array **(n1, n2, n3, n4, n5, n6, n7, n8, n9, n10)** may be printed as a tree

```
              n1
            n2,  n3
         n4,  n5,  n6,  n7,
    n8,  n9,  n10
```

using an appropriate indentation to show the layers. Your task is to decide what indentation provides a good enough readability of the printed tree, and to find out how to implement it.

Add these new public member functions to the **your_name_Priority_queue.cpp** file. Don’t forget to add comments (header comment and comments explaining your code).

Using **priority_queue.cpp** as a template, create a file **your_name_Priority_queue.cpp** with a **`main()`** function that performs the following tests.

a) Test for an empty queue.

b) Test for a queue containing just one element.

c) Test for a queue with a small number of random elements (3 - 5 elements).

d) Test for a queue with approx. 70 random elements to print more substantial tree.

Each test case should appear in your **main()** function separately and should have the following structure.

- A comment explaining the test case.
- Declare a variable of type **Priority_queue**.
- Populate the queue with elements using **enqueue()** method.
- Show the results of calling each of your new methods on the queue by printing the contents of the queue before and after the operation using the **show()** and **print_visual()** methods.

Don’t copy tests for existing member functions, but rather populate the main() function

with tests demonstrating your new functions.

The header comments in your source files are mandatory, and should contain at least the following:

Your name;
Date;
File name;
Short description of the file contents (tasks, purpose, etc.);
Platform on which it has been designed (Mac OS, Windows, etc.).

Submit the source code of your solution (file **your_name_Priority_queue.cpp**) electronically to maugusto@nps.edu

The deadline is **Monday, December 11, 2017, midnight.**