Maximum possible marks: 100 Points

Due date:
11:59pm, Sunday 10/27/2019 for Monday lab.
11:59pm, Tuesday 10/29/2019 for Wednesday labs.
11:59pm, Thursday 10/31/2019 for Friday lab.

Purpose:
For this lab, you will implement a Minmax heap in C++.

General Requirements:
In this assignment, you will develop an array-based implementation of a Minmax heap. The initial build of the Minmax heap should use the top-down approach. No duplicates are allowed to be inserted. Also, each time you insert/remove an element, you should organize the heap following the minmax heap property.

In the Minmax heap:
The root of T will be at the minimum level and the next level should be a maximum level. Minimum and maximum levels will alternate until all the integers in the data.txt file are inserted into the heap.

Here is where the min and max nodes are located:
- min nodes: even levels (0, 2, 4...)
- max nodes: odd levels (1, 3, 5...)

The Minmax heap methods should be implemented as follows:

- BuildHeap() - should build the Minmax heap using the top-down approach.
- Insert(x) - should insert x into the Minmax heap. This means you should add a new element into the heap. After the insertion of new element, the heap should still satisfy the MinMax heap property.
- Delete(x) - should delete the element x from the Minmax heap. After deleting the element x, the heap should still satisfy the Minmax heap property.
- MinLevelElements() - should print out the elements of the Minmax heap at even levels in level order.
- MaxLevelElements() - should print out the elements of the Minmax heap at odd levels in level order.
- Exit() – should exit from the program.

The file you will read the processes from will be data.txt. You are allowed to hard code the file name in your program.
Let's look at how the Minmax Heap works diagrammatically.

**Step 1:** Insert 10

```
10
```

**Step 2:** Insert 11

```
10
  11
```

**Step 3:** Insert 5

```
10
  11
   5
```

Here the Minmax Heap property is violated. 10 is not smaller than 5. So swap the 5 with the 10.

Note: The element at the root will always be the smallest of the elements.

**Step 4:** Satisfying the heap property

```
  5
 /   \
11   10
```
Step 5: Insert 13

When 13 is inserted to the left child of 11, 13 will be greater than 11. So 13 and 11 will interchange their positions. The diagram above is the heap after 13 is inserted.

Step 6: Insert 19

Now when 19 is inserted as the right child of 13, 19 is greater than 13. So 13 and 19 will be swapped. The below diagram is the representation of the heap after 19 is added and swapped with 13.
Final Step: The below diagram will be the final Minmax Heap representation after all the elements are inserted and heapify operations are performed until the Minmax property is satisfied.

In this lab, you should build the heap using the samples which are in data.txt. After that, your program should have a simple menu like this:

------------------------------------------------------------
Please choose one of the following commands:
  1- BuildHeap
  2- Insert
  3- Delete
  4- MinLevelElements
  5- MaxLevelElements
  6- Exit

>Enter your choice:
>1
> Output: 2,
   25, 22,
   8, 5, 10, 9,
   11, 19, 13, 7

------------------------------------------------------------
Please choose one of the following commands:
  1- BuildHeap
  2- Insert
3- Delete
4- MinLevelElements
5- MaxLevelElements
6- Exit

>Enter your choice:
>2
>Enter the element to be inserted: 23
>Output: 23 has been inserted successfully.

Please choose one of the following commands:
1- BuildHeap
2- Insert
3- Delete
4- MinLevelElements
5- MaxLevelElements
6- Exit

>Enter your choice:
>5
>Output: 25, 22,
  11, 19, 13, 7, 23

Please choose one of the following commands:
1- BuildHeap
2- Insert
3- Delete
4- MinLevelElements
5- MaxLevelElements
6- Exit

>Enter your choice:
>3
>Enter element to be deleted: 1
>Output: 1 has been deleted successfully.

Please choose one of the following commands:
1- BuildHeap
2- Insert  
3- Delete  
4- MinLevelElements  
5- MaxLevelElements  
6- Exit  

>Enter your choice: 
>4  
> Output: 2, 8, 5, 10, 9  

-----------------------------

Please choose one of the following commands:  
1- BuildHeap  
2- Insert  
3- Delete  
4- MinLevelElements  
5- MaxLevelElements  
6- Exit  

>Enter your choice: 
>6  
> Output: Bye!!!  

Questions:  
Please answer the following question in not more than 5 lines each and submit it with your implemented code in PDF format.  
1. Explain a situation where you would implement Minmax Heap using a real world example.  

Submission:  
Follow the conventions below to facilitate grading:  

Report  
• Please include your answer (answer.pdf) in your folder before compressing it.  

Source Code  
Place all your source files (*.cpp, *.hpp) and input files in a single folder with no subfolders.  
• Name your folder using the convention Lastname_Lab07 (e.g., Smith_Lab07).  
• Include a functioning Makefile inside the folder. (The makefile should also include the clean command.)  
• Verify that your code runs on the lab Linux machines before submission.
Compressed File

- Compress using .zip, .rar, or .tar.gz.
- Name your file using the convention Lastname_Lab07 (e.g., Smith_Lab07.zip).

Email

- Use the following subject for your email: Lastname_Lab07 (e.g., Smith_Lab07).
- Send your code to chiranjeevi.pippalla@ku.edu if you are in one of Chiru’s sections or to anubhav@ku.edu if you are in one of Anubhav’s sections.
- Anytime you have a question about the lab, put the word question in the subject of the email.