**COMP 203 Data Structures and Algorithms, Fall 2024**

**Lab Assignment 11**

**Deadline: 23.12.2024 11:00 am**

**Read the questions and rules carefully. They are clear and well defined.**

**Rules:**

1. **No Cheating:** You are not allowed to collaborate with your friends and use any kind of websites or AI. If your homework gives a sign of any of them, **directly it will be graded as zero**.
2. **Goal:** Please do your homework alone. Our main aim is to **learn**.
3. **Submission:** Submit your work in **one java files.** **DON’T USE ZIP/RAR etc. In these cases, your points will be deducted by 30%.**
4. **Coding policy:** Explain your code in comments. **This is a must!**
5. **Latency policy:** A 30% deduction will be applied for each day of late submission.

**Files to submit:** LLPQ.java

**Linked List Implementation of Priority Queue**

Implement Priority Queue abstract data structure using Node data structure. You can construct Priority Queue data structure similar to Singly Linked List but the rules of Priority Queue is different as you know. **You will use the sorted list implementation**. It means that you will add the nodes in priority queue according to their priority.

**Example:** Your priority queue will be as follows: head-> (A,1)->(C,2) -> (F,3) -> (D,4) ->null. Here the head node has (A,1) and has the highest priority.

**Implement the following classes and methods in java: (100pt)**

1. Implement Node<char, int> class for Priority Queue with its constructor with the parameter char data and int priority. (5pt**)**

**Hint:** Each node has a char data and int priority and Node next variables

1. Implement LLPQ class with the class variables and a constructor. You should have *int size* variable to keep track of size of the priority queue. (10pt)
2. Insert(char addedData, int priority ) that inserts the node with the given value *addedData and* given *priority* into the priority queue. (20pt)
3. RemoveMin() that removes the element in the priority queue with the highest priority and the returns the removed node’s data. (20pt)
4. Min() returns the value of the node having the highest priority. (10pt)
5. Size() that returns the number of nodes in the priority queue. (5pt)
6. PrintQueue that prints the data in the priority queue. (10pt)
7. Test your methods in the main by creating the following priority queue. (Creating the given priority queue:10 pt, testing 5 methods: 5x2=10 pt)

Head-> (A,1)-> (G,2)->(U,3)->(C,4)->(E,5)->NULL

Include comments of your code for each method and class.