Subject #2E | 15.06.2022

Implement a C application for managing the queue of processes that need to be executed by a CPU. For this, you can use a Queue structure that handles processes with the following structure: process name (char*), priority (unsigned short), needed memory (unsigned int), and execution time (unsigned short).

1. Print all the processes in the queue without changing the queue structure and only by using queue-specific operations. **0.5p**
2. Process ***only*** the processes with a specific priority (value given as a parameter to the function); processing a process means removing it from the initial structure, displaying it at the console, and releasing the occupied memory at the end. The ones that do not meet the criteria will be kept in the queue. **1p**
3. Extract all the processes and save them into a circular simple linked list by inserting them in descending order by the amount of memory that they need. **1p (-0.5 if it is not ordered)**
4. Determine which are the processes from the newly created list that have the lowest resource consumption in terms of memory and time and display them to the console. **1p**
5. Delete the processes from the previous list that have the needed memory between [base-x; base+x]; *base* and *x* are two given parameters to the function; **1p**
6. Split the previous list into two separate lists based on a given process name, so that the first list will include all the processes from the start of the list up to the given process and the second list will include the rest of the processes; print the two newly created lists; **1.5p**

The following items should be considered for the implementation:

- ***Projects with compilation issues are NOT going to be evaluated;***
- ***Functions that are not tested in the main() function are not taken into account at evaluation;***
- ***Source code that is commented is NOT going to be evaluated;***