CS200 Introduction to Programming
Spring 2022
Programming Assignment 2

Guidelines

Rules:

- The object is not simply to get the job done, but to get it done in the way that is asked for.
- Any cheating case will be reported to Disciplinary Committee without any delay.

Coding Conventions:

- Constants are “ALLCAPS” or “ALL_CAPS”.
- Variables are “allsmall” or “all_small”.
- All function names must be “firstWordSmallAllOtherWordsCamelCase”.
- All class names must be “CamelCaseWords”.
- All curly brackets defining a block must be vertically aligned.
- Use tabs where needed instead of spaces.
- File naming: RollNumber_PA2.zip

Course Learning Objective (CLOs):

- CLO1: Enabling Knowledge:
    - (C1) use object-oriented programming model: abstract data types, encapsulation, inheritance, and polymorphism to code algorithmic solutions using standard coding conventions.
    - (C1) use of fundamental features of an object-oriented language like C++.
- CLO2: Critical Thinking and Analysis:
    - (C4) analyze the requirements for solving simple algorithmic problems.
- CLO3: Problem Solving:
    - (C6) design algorithm and implement program code in an object-oriented programming language such as C++ to solve simple algorithmic computing problems, based on the analysis of the requirements.
    - (C5) evaluate the correctness of the proposed solution.

- CLO4: Communication:
    - (C2) explain key concepts of algorithmic design in written form.
- CLO5: Responsibility:
    - (C3) apply relevant conventions, standards, and ethical considerations to writing computer programs.

Total Marks for the assignment: 100

Q1) Total Marks: 70

In this question, you will implement a **To-Do List** using a recursive singly linked list. The node will be a structure called ‘Node’ which has 4 variables: [5]

- Task
- Time Object: You need to implement a *Time* class that follows 24-hour format which will have the following attributes: [5]
    - Private Data Members: Hour, Minute
    - Setters and Getters
    - Overloaded “<<” operator using friend function.
- Time needed to complete the task (in minutes)
- Pointer to next node

Your To-Do List will be a linked list of these nodes. You will need to write functions to:

- Insert a Task at the end (recursive definition) [5]
- Insert a Task after a specific Task (iterative definition) [5]
- Delete a Task (from the end) (recursive definition) [5]
- Display the To-Do List (recursive definition). To display the time of the task you must use << operator. [5]
- Find the task which needs the lowest amount of time (recursive definition) [10]
- Determine the length of the To-Do List (recursive definition) [5]
- Search for a task in the To-Do List (search by task description, search by Time needed to complete the task) (recursive definition) [5]
- Delete the To-Do List (recursive definition) [10]

You must implement the functions iteratively or recursively as mentioned above.

You are not allowed to make Inline Functions.

Make the appropriate prompts to the user for entering the details of the task where required and also to determine whether the user wants to search for a task by its description or time needed.

In the main function, you will create a menu that will allow the user to avail of any of these functionalities and will continue doing so until the user desires to stop.
[10]

Ensure that your program makes the appropriate checks where required so that your program never crashes unexpectedly.
[5]

Q2) Total Marks: 30

In this Question, you must implement the same To-Do List from question 1, only this time you will have to implement the To-Do list in a ***Doubly Linked List***.

Here you do not have to implement all the functions we have asked for in question 1, instead you will be doing insertion sort (Pseudocode given at the end) based on Time needed to complete the task. [10]

Your function should return you a separate sorted linked list. You can assume that the user has tasks for only a single day in their To-Do List.

In this question you also need to implement:

- A function that inserts at the end of the list [2]
- A function that inserts at the start of the list [2]
- A function that deletes from the end of the list [2]
- A function that deletes from the start of the list [2]
- A recursive print function to display your To-Do List. [2]

In your main function make a menu that takes input from the user to do any of the above-mentioned functions. The program exits when the user enters "stop" as input. [5]

Submission Guidelines:

Submit *rollNumber_PA2.zip* which contains:

- RollNumber_PA2_Q1.cpp
- RollNumber_PA2_Q2.cpp

Following coding conventions and submission guidelines carries 5 marks throughout the assignment. [5]

You can declare and define appropriate constructors, destructor, setters, getters, and other helper functions if needed.

Pseudocode for insertion sort:

```
A is an unsorted array

i ← 1
while i < length(A)
 j ← i
 while j > 0 and A[j-1] > A[j]
 swap A[j] and A[j-1]
 j ← j - 1
 end while
 i ← i + 1
end while
```