CS-2001 Data Structures (Fall 2023) Assignment #02

"ProjectSchedPro: Integrated Project Scheduling System"

Objective:
The objective of this assignment is to create a project scheduler that efficiently plan and manage tasks, resources, and timelines, ensuring project completion within the set timeframe. It involves optimizing resource allocation and sequencing tasks logically for successful project delivery.

Assignment Statement:
In the realm of Project Management, one of the fundamental tasks is scheduling, where we determine the timeline for project completion. Traditionally, this involves compiling a list of tasks, their respective durations, and dependencies among them. One widely-used method for scheduling is the Critical Path Method (CPM). However, we aim to enhance this approach by taking into account not only task-related information but also the availability of resources and their skill sets when generating the project schedule.

Let's consider the key structures for our system:

Task: Each task is characterized by its unique identifier (id), duration, start time, dependencies on other tasks, and the skill required to execute it (skill_id).

Project: A project consists of a unique identifier (id), duration, and a collection of tasks.

Skill: Skills are associated with specific identifiers (id) and proficiency levels.

Resource: Resources are defined by their identifiers (id), availability, and the skill they possess.

(Note: You can always add additional data members if you feel it will result in better/efficient design).

A few essential points to note:

The start time for **at least one task** in a project is set to zero (0).

A project comprises N tasks and involves the allocation of resources from a pool of R available resources. Both N and R will be input by the user.

Remember the first task entered does not necessarily need to have start time zero.

Now, let's outline the functions we need to implement for this project, keeping in mind the necessary parameters and return types:

addResources(); //Add resources to the project, and their skill and proficiency.

addTasks(); //Add tasks to the project, and their relevant information.

setTaskDuration(); //Modify the duration of all tasks in the project.

set_nth_TaskDuration(); //Change the duration of a specific task.

printTaskDependencyList(); //Display the dependencies of all task.

calculateBasicSchedule(); //Utilize the Critical Path Method to compute and display the project's schedule and print the completion time.

```
Basic schedule:
ID        ES        EF        LS        LF        Slac
0         0         1         0         1         0
1         0         2         2         4         2
2         1         4         1         4         0
```

(Example for schedule)

printCriticalTasks(); //Identify and return the critical tasks while displaying them. The sum of their durations should match the project's completion time.

completionTimeWithResources(); //Calculate the project's completion time considering resource availability, allowing tasks to start based on resource availability. Display the schedule as well.

Keep in mind the skill for resource and task must be the same for resource allocation to task.

```
Resource schedule:
ID        ES        EF        Slac      Res
0         0         1         0         0
1         1         3         2         0
2         1         4         0         1
```

(Example for schedule)

completionTimeWithResourceProficiency(); //Compute the project's completion time, taking into account resource availability and their proficiency levels, enabling tasks to commence accordingly. Display the schedule as well.

(Can follow same example for resource schedule for its schedule)

Other important points will be a **Menu, validation of all kinds, comments, and a Readme** containing your code explanation and references used.

By implementing these functions effectively, we can create a robust project scheduling system that integrates task details, resource allocation, and skill proficiency to optimize project completion timelines.

Some links have been provided but for more details regarding resource allocation there is a lot of information available on the net, just remember to include these links in Readme.

(https://asana.com/resources/critical-path-method)

https://www.geeksforgeeks.org/software-engineering-critical-path-method/

Important notice: DO NOT USE ARRAYS OR BUILT-IN DATA STRUCTURES, use **Linked Lists** and its variations.

Instructions (before starting the assignment):
1. Assignments are to be done individually.

2. The code you write must be your own and you must understand each part of your code. You are encouraged to get help from the course instructors through google classroom and email.

3. Apply all validations for invalid inputs.

4. Plagiarism: Plagiarism of any kind (copying from others, copying from the internet, etc.) is not allowed. If found plagiarized, you will be awarded zero marks in the assignment. Repeating such an act can lead to strict disciplinary actions and failure in the course.

5. Please start early otherwise you will struggle with the assignment.

Submission Guidelines

a. submit both .h and .cpp file Your submission must contain your
name, student-id, and assignment # on the top of the file in the comments. Example
the first line of assignment you should write
//Maheen_Arshad_22i111. Missing this will result in 20% marks deduction in each question.

b. Move your .cpp and .h file in one folder. The folder must contain only submission.cpp and h files(no binaries, no exe files etc.,). If we are unable to download your submission due to any reason you will be awarded zero mark.

c. Run and test your program on machine before submission. If there is a syntax error, zero marks will be awarded in that specific question.

d. Rename the folder as ROLL-NUM_SECTION (e.g. 21i-0001_A) and compress the folder as a **zip file**. (e.g. 21i-0001_A.zip). Only **zip file** will be acceptable.

e. Submit the .zip file on Google Classroom within the deadline. Late submission will be marked zero. No exceptions

f. Submission other than Google classroom (e.g. email etc.) will not be accepted.

g. The student is solely responsible to check the final zip files for issues like corrupt files, viruses in the file, mistakenly exe sent. If we cannot download the file from Google classroom due to any reason it will lead to zero marks in the assignment.