**Critical Path Method (CPM) Project**

**Overview**

This project is a CPM (Critical Path Method) application designed to assist in project management, task scheduling, and resource allocation. The program allows you to create a project, add resources, define tasks, specify task dependencies, and calculate schedules.

**Features**

**1. Adding Resources**

* Use the addResources function to add resources to your project. You can provide information about each resource, including their skills and proficiency levels.

**2. Adding Tasks**

* The addTasks function lets you add tasks to your project. You can provide task descriptions and their durations. The first task doesn't have to be a starting task.

**3. Modifying Task Durations**

* You can modify the duration of all tasks in the project using the setTaskDuration function. Additionally, the set\_nth\_TaskDuration function allows you to change the duration of a specific task.

**4. Task Dependencies**

* The printTaskDependencyList function displays the dependencies of all tasks, ensuring correct formatting and readability.

**5. Basic Schedule Calculation**

* The calculateBasicSchedule function calculates the project's schedule, including completion times, early times, and late times. Critical tasks are identified.

**6. Critical Tasks Identification**

* The printCriticalTasks function identifies and returns critical tasks, along with the sum of their durations.

**7. Completion Time with Resources**

* The completionTimeWithResources function calculates the project's completion time, considering resource availability, task dependencies, and resource priority. It displays the schedule.

**8. Completion Time with Resource Proficiency**

* The completionTimeWithResourceProficiency function computes the project's completion time, considering resource availability and proficiency levels. It efficiently allocates resources based on their skills.

**User Menu**

The program offers a user-friendly menu for interacting with the system, including options to add resources, tasks, modify durations, view dependencies, and calculate schedules.

**Input Validation**

**To** maintain data integrity, the program validates all user inputs, providing clear error messages for invalid inputs.

**Usage**

1. Run the main.cpp file to launch the program.
2. Give inputs according to instructions by the menu.

**Code Structure and Organization**

The code is organized into functions and classes for clarity and modularity. It adheres to good coding practices to ensure efficient execution.

**Dependencies**

This project does not have external dependencies other than the standard C++ libraries.

**Contribution**

Contributions are welcome! If you'd like to improve the program or fix issues, feel free to submit pull requests.

**Credits**

This project was created by Yahya Hassan 22i-1549, as Assignment 2 of Data structures.

**References**

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

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