Name - Srushti Bhivaji Salgar
PRN - B24CE1079
Class - SE 2 Batch - A
Subject - Data Structures
Assignment 4

/*PROBLEM STATEMENT:
Simple Task Scheduler:
Write a program that implements a simple task scheduler using a singly linked list. Each node in the linked list represents a task with its priority and execution time. Tasks are scheduled based on their priority, with higher priority tasks being executed first.*/

CODE

```
#include <iostream>
using namespace std;

class Node {
public:
    string task_name;
    int priority;
    int exe_time;
    Node* next;

    Node(string tn, int p, int e) {
        task_name = tn;
        priority = p;
        exe_time = e;
        next = NULL;
    }

    void display() {
        cout << "Task: " << task_name << endl;
        cout << "Priority Level: " << priority << endl;
        cout << "Execution Duration: " << exe_time << " sec" << endl;
        cout << "---------------------------------" << endl;
    }
};

int main() {
    Node* header = NULL;
    Node* prev = NULL;
    Node* current = NULL;
```

```
Node* temp = NULL;

int n;
string tn;
int p;
int e;

cout << "Enter total number of tasks to schedule: ";
cin >> n;

for (int i = 0; i < n; i++) {
    cout << "\n--- Task " << i + 1 << " ---" << endl;
    cout << "Enter Name: ";
    cin >> tn;
    cout << "Enter Priority (bigger number = higher priority): ";
    cin >> p;
    cout << "Enter Time Required (in sec): ";
    cin >> e;

    temp = new Node(tn, p, e);

    if (header == NULL) {
        header = temp;
    } else {
        if (header->priority < temp->priority) {
            temp->next = header;
            header = temp;
        } else {
            prev = header;
            current = prev->next;
            while (current != NULL && current->priority >= temp->priority) {
                prev = current;
                current = current->next;
            }
            prev->next = temp;
            temp->next = current;
        }
    }
}

cout << "\n------------------------------" << endl;
cout << " Final Task Execution Order:" << endl;
cout << "------------------------------\n" << endl;
```

```
    Node* t = header;
    while (t != NULL) {
        t->display();
        t = t->next;
    }

    return 0;
}
```

OUTPUT

```
Enter total number of tasks to schedule: 2

--- Task 1 ---
Enter Name: t1
Enter Priority (bigger number = higher priority): 4
Enter Time Required (in sec): 10

--- Task 2 ---
Enter Name: t2
Enter Priority (bigger number = higher priority): 6
Enter Time Required (in sec): 24

------------------------------
 Final Task Execution Order:
------------------------------

Task: t2
Priority Level: 6
Execution Duration: 24 sec
--------------------------------
Task: t1
Priority Level: 4
Execution Duration: 10 sec
--------------------------------
```