ASSIGNMENT-4

```
/*Name: Sahil Badve PRN: B24CE1114 Div: S.Y.B-Tech 2 Batch C*/
//Simple Task Scheduler

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

// Node class
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;
    }
};

// TaskScheduler class implementing singly linked list
class TaskScheduler {
private:
    Node* head;

public:
    TaskScheduler() {
        head = NULL;
    }

    // Function to create and add tasks
    void create() {
        char ch;
        do {
            Node* newnode = getnode();   // create new node
            insert_node(newnode);        // insert it in correct place
            cout << "Do you want to add more tasks? (y/n): ";
            cin >> ch;
        } while (ch == 'y' || ch == 'Y');
```

```
}

// Get details from user and create node
Node* getnode() {
    string tn;
    int p, e;
    cout << "Enter task name: ";
    cin >> tn;
    cout << "Enter priority: ";
    cin >> p;
    cout << "Enter execution time: ";
    cin >> e;

    Node* newnode = new Node(tn, p, e);
    return newnode;
}

// Insert node in sorted order (based on priority)
void insert_node(Node* newnode) {
    if (head == NULL || newnode->priority > head->priority) {
        newnode->next = head;
        head = newnode;
    } else {
        Node* temp = head;
        while (temp->next != NULL && temp->next->priority >= newnode->priority) {
            temp = temp->next;
        }
        newnode->next = temp->next;
        temp->next = newnode;
    }
}

// Display all tasks
void display() {
    if (head == NULL) {
        cout << "No tasks scheduled!\n";
        return;
    }
    cout << "\nScheduled Tasks:\n";
    Node* temp = head;
    while (temp != NULL) {
        cout << "Task: " << temp->task_name
             << " | Priority: " << temp->priority
             << " | Execution Time: " << temp->exe_time << "s\n";
```

```
            temp = temp->next;
        }
    }

    // Execute (delete) tasks in order of priority
    void delete_node() {
        if (head == NULL) {
            cout << "No tasks to execute!\n";
            return;
        }

        cout << "\nExecuting Tasks: \n";
        while (head != NULL) {
            cout << "Executing '" << head->task_name
                 << "' (Time: " << head->exe_time << "s)\n";
            Node* temp = head;
            head = head->next;
            delete temp;
        }
        cout << "All tasks executed!\n";
    }
};

// Main Function
int main() {
    TaskScheduler scheduler;
    int choice;

    do {
        cout << "\n===== Simple Task Scheduler =====\n";
        cout << "1. Add Tasks\n";
        cout << "2. Display Tasks\n";
        cout << "3. Execute All Tasks\n";
        cout << "4. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
        case 1:
            scheduler.create();
            break;
        case 2:
            scheduler.display();
            break;
```

```
        case 3:
            scheduler.delete_node();
            break;
        case 4:
            cout << "Thankyou for using task scheduler\n";
            break;
        default:
            cout << "Invalid choice! Try again.\n";
        }
    } while (choice != 4);

    return 0;
}
```

OUTPUT:-

—-- Simple Task Scheduler —--
1. Add Tasks
2. Display Tasks
3. Execute All Tasks
4. Exit
Enter your choice: 1
Enter task name: abc
Enter priority: 50
Enter execution time: 20
Do you want to add more tasks? (y/n): y
Enter task name: def
Enter priority: 90
Enter execution time: 50
Do you want to add more tasks? (y/n): y
Enter task name: ghi
Enter priority: 70
Enter execution time: 30
Do you want to add more tasks? (y/n): n

—-- Simple Task Scheduler —--
1. Add Tasks
2. Display Tasks
3. Execute All Tasks
4. Exit

Enter your choice: 2

Scheduled Tasks:
Task: def | Priority: 90 | Execution Time: 50s
Task: ghi | Priority: 70 | Execution Time: 30s
Task: abc | Priority: 50 | Execution Time: 20s

—-- Simple Task Scheduler —--
1. Add Tasks
2. Display Tasks
3. Execute All Tasks
4. Exit
Enter your choice: 3

Executing Tasks:
Executing 'def' (Time: 50s)
Executing 'ghi' (Time: 30s)
Executing 'abc' (Time: 20s)
All tasks executed!

—-- Simple Task Scheduler —--
1. Add Tasks
2. Display Tasks
3. Execute All Tasks
4. Exit
Enter your choice: 4
Thankyou for using task scheduler