ASSIGNMENT NO:4
Load Balancing

Name: Munajja Mujafar Dalimbkar
Class: B Tech II
PRN No.: B25CE2011

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;

 {
 string name;
int priority;
int execTime;
 Task* next;
 };
{

 Task* head;
 public:
TaskScheduler()
{
 head = nullptr;
}
void addTask(string taskName, int priority, int execTime)
{
Task* newTask = new Task
{
taskName, priority, execTime, nullptr};
if (head == nullptr || head->priority < priority)
{
```

```
newTask->next = head;
 head = newTask;
}
 else
{
 Task* temp = head;
while (temp->next != nullptr && temp->next->priority >= priority)
{
 temp = temp->next;
 }
newTask->next = temp->next;
 temp->next = newTask;
}
cout << "Task \"" << taskName << "\" added with Priority " << priority << " and Execution Time
" << execTime << "s\n";
}
void executeTask()
 {
if (head == nullptr)
{
cout << "No tasks to execute.\n";
 Return;
 }
Task* temp = head;
cout << "Executing Task: " << temp->name << " | Priority: " << temp->priority << " | Execution
Time: " << temp->execTime << "s\n";
 head = head->next;
 delete temp;
}
void displayTasks()
{
if (head == nullptr)
{
 cout << "No tasks scheduled.\n"; return;
 }
cout << "\nScheduled Tasks (Higher priority first):\n";
cout << "----------------------------------------\n";
Task* temp = head;
while (temp != nullptr)
 {
```

```
cout << "Task: " << temp->name << " | Priority: " << temp->priority << " | Time: " <<
temp->execTime << "s\n";
 temp = temp->next;
 }
cout << "---------------------------------------\n";
 }
 };
 int main()
{
TaskScheduler scheduler;
 int choice, priority, time;
string name;
 char ch = 'Y';
do
{
cout << "\n--- Simple Task Scheduler ---\n";
 cout << "1. Add Task\n";
 cout << "2. Execute Next Task\n";
cout << "3. Display All Tasks\n";
cout << "4. Exit\n";
 cout << "Enter your choice: ";
 cin >> choice;
 switch (choice)
{
case 1:
cout << "Enter Task Name: ";
cin >> name;
 cout << "Enter Priority (higher = more important): ";
cin >> priority;
cout << "Enter Execution Time (in sec): ";
cin >> time;
 scheduler.addTask(name, priority, time);
Break;
 case 2:
scheduler.executeTask();
break;
case 3:
 scheduler.displayTasks();
 break;
case 4:
```

```
cout << "Exiting program...\n";
return 0;
 Default:
 cout << "Invalid choice! Try again.\n";
}
cout << "Do you want to continue? (Y/N): ";
 cin >> ch;
 }
 while (ch == 'Y' || ch == 'y');
return 0;
 }
```

Output:

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

Enter Task Name: Backup
Enter Priority (higher = more important): 5
Enter Execution Time (in sec): 10
Task "Backup" added with Priority 5 and Execution Time 10s

Do you want to continue? (Y/N): Y

Enter your choice: 1
Enter Task Name: Update
Enter Priority (higher = more important): 2
Enter Execution Time (in sec): 5
Task "Update" added with Priority 2 and Execution Time 5s

Do you want to continue? (Y/N): Y

Enter your choice: 3
Scheduled Tasks (Higher priority first):

Task: Backup | Priority: 5 | Time: 10s
Task: Update | Priority: 2 | Time: 5s
