 **Data Structures Experiment 4**



**Name** : [Anurag Patil](mailto:anuragpatil2024.comp@mmcoe.edu.in)

**PRN** : B24CE1047   
**Branch and Batch** : Computer Engineering SY1 Batch (B)



**Title** : Simple Task Scheduler  
**Problem Statement** :

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>

#include <string>

using namespace std;

class Node {

public:

string name;

int priority;

int execution\_time;

Node\* next;

Node(string name, int priority, int execution\_time) {

this->name = name;

this->priority = priority;

this->execution\_time = execution\_time;

this->next = nullptr;

}

};

void input\_Task(Node\*& head, Node\* newTask) {

 if (head == nullptr || newTask->priority > head->priority) {

newTask->next = head;

head = newTask;

return;

}

Node\* current = head;

while (current->next != nullptr && current->next->priority >= newTask->priority) {

current = current->next;

}

newTask->next = current->next;

current->next = newTask;

}

void display\_task(Node\* head) {

cout << "\nExecution Tasks:\n";

Node\* temp = head;

while (temp != nullptr) {

cout << "Task name: " << temp->name

<< " [Priority: " << temp->priority << "] "

<< "Execution Time: " << temp->execution\_time << " ms\n";

temp = temp->next;

}

}

void execute\_task(Node\* head) {

cout << "\nExecuting Tasks:\n";

Node\* temp = head;

while (temp != nullptr) {

cout << "Executing Task '" << temp->name

<< "' [Priority: " << temp->priority

<< "] for " << temp->execution\_time << " ms...\n";

temp = temp->next;

}

cout << "All tasks executed.\n";

}

int main() {

int n;

cout << "Enter number of tasks to schedule: ";

cin >> n;

Node\* head = nullptr;

for (int i = 0; i < n; i++) {

string name;

int priority;

int execution\_time;

cout << "\nTask " << i + 1 << " Name: ";

cin >> name;

cout << "Priority (higher = more important): ";

cin >> priority;

cout << "Execution Time (ms): ";

cin >> execution\_time;

Node\* newTask = new Node(name, priority, execution\_time);

input\_Task(head, newTask);

}

display\_task(head);

execute\_task(head);

Node\* temp;

while (head != nullptr) {

temp = head;

head = head->next;

delete temp;

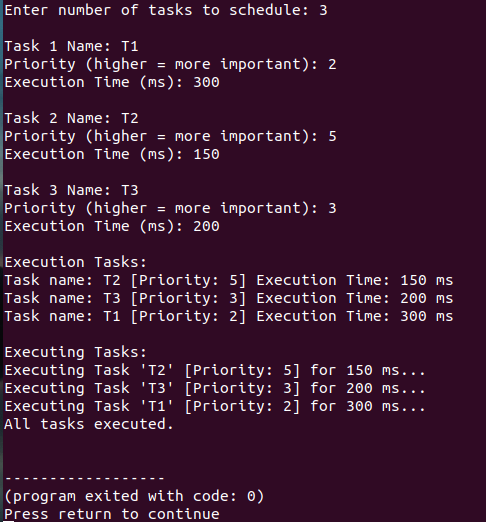
}

return 0;

}

**OUTPUT :**







Thank You !!!