DS Assignment 4
Name :- Prince Vyas
PRN :- B24CE1062

Program :-

```
#include <iostream>
#include <string>
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 Name: " << task_name << endl;
        cout << "Priority: " << priority << endl;
        cout << "Execution Time: " << exe_time << " seconds" << 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 << "B24CE1062-PRINCE VYAS" << endl;
cout << "How many tasks do you want to add? ";
cin >> n;
cin.ignore(); // To clear newline character from input buffer after reading n

for (int i = 0; i < n; i++) {
    cout << "Enter task name: ";
    getline(cin, tn);  // Use getline to allow spaces in task name
    cout << "Enter task priority (higher number means higher priority): ";
    cin >> p;
    cout << "Enter execution time (in seconds): ";
    cin >> e;
    cin.ignore(); // Clear newline after reading integer inputs

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

    // Insert into the priority linked list
    if (header == NULL || header->priority < temp->priority) {
        // Insert at the front if list empty or temp has higher priority than header
        temp->next = header;
        header = temp;
    } else {
        prev = header;
        current = header->next;
        // Traverse until we find a node with lower priority than temp
        while (current != NULL && current->priority >= temp->priority) {
            prev = current;
            current = current->next;
        }
        // Insert temp between prev and current
        prev->next = temp;
```

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

    cout << "\n------------------------------------------------\n" << endl;
    cout << "The Priority List is as follows\n" << endl;

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

    // Clean up memory
    t = header;
    while (t != NULL) {
        Node* toDelete = t;
        t = t->next;
        delete toDelete;
    }

    return 0;
}
```

OUTPUT:-

```
B24CE1062-PRINCE VYAS
How many tasks do you want to add? 4
Enter task name: ds
Enter task priority (higher number means higher priority): 1
Enter execution time (in seconds): 12
Enter task name: oop
Enter task priority (higher number means higher priority): 2
Enter execution time (in seconds): 50
Enter task name: DECO
Enter task priority (higher number means higher priority): 3
Enter execution time (in seconds): 16
Enter task name: HVPE
Enter task priority (higher number means higher priority): 4
Enter execution time (in seconds): 56
```

```
------------------------------------------

The Priority List is as follows

Task Name: HVPE
Priority: 4
Execution Time: 56 seconds
-----------------------------
Task Name: DECO
Priority: 3
Execution Time: 16 seconds
-----------------------------
Task Name: oop
Priority: 2
Execution Time: 50 seconds
-----------------------------
Task Name: ds
Priority: 1
Execution Time: 12 seconds
-----------------------------
```