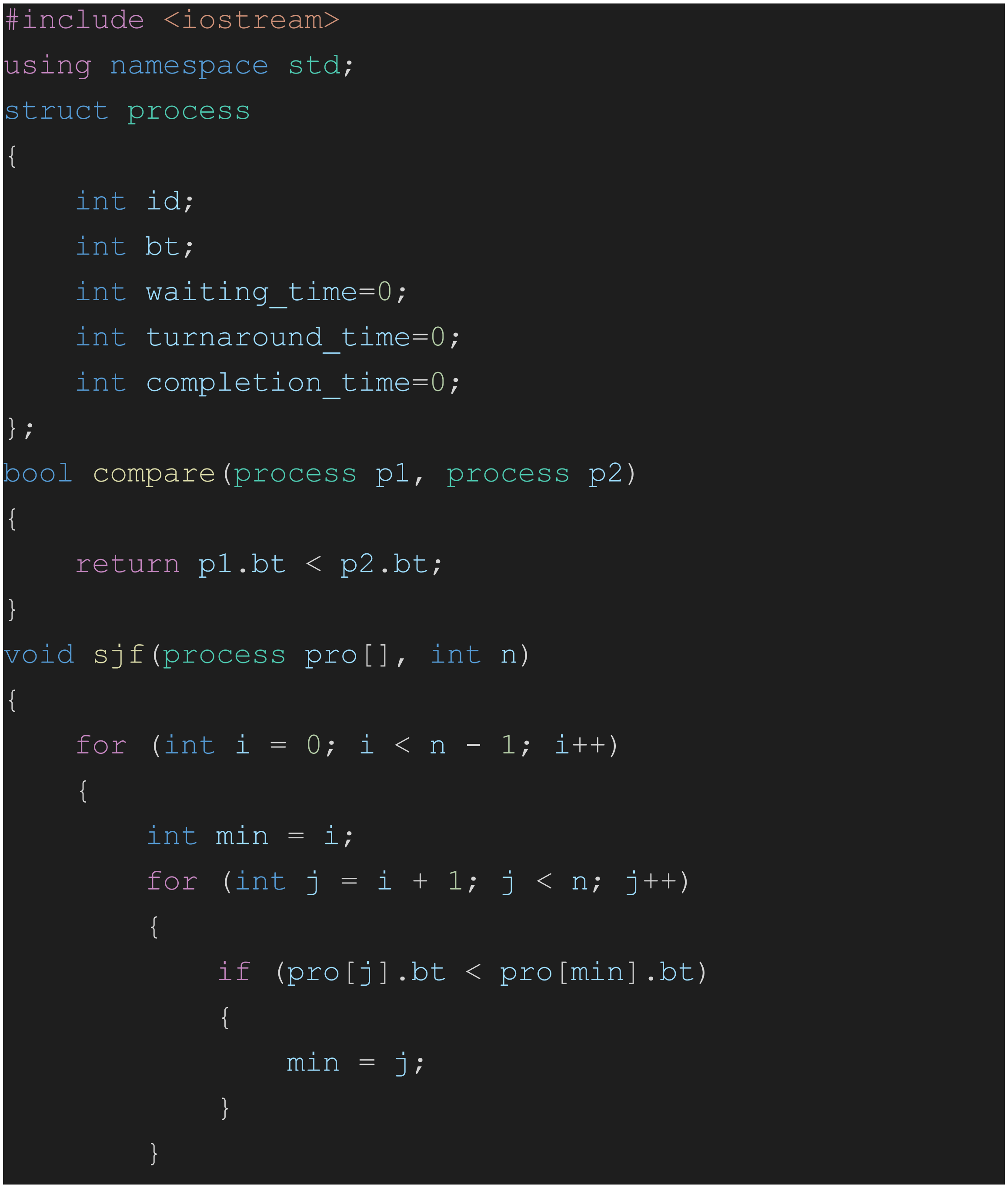
OS ASSIGNMENT-7

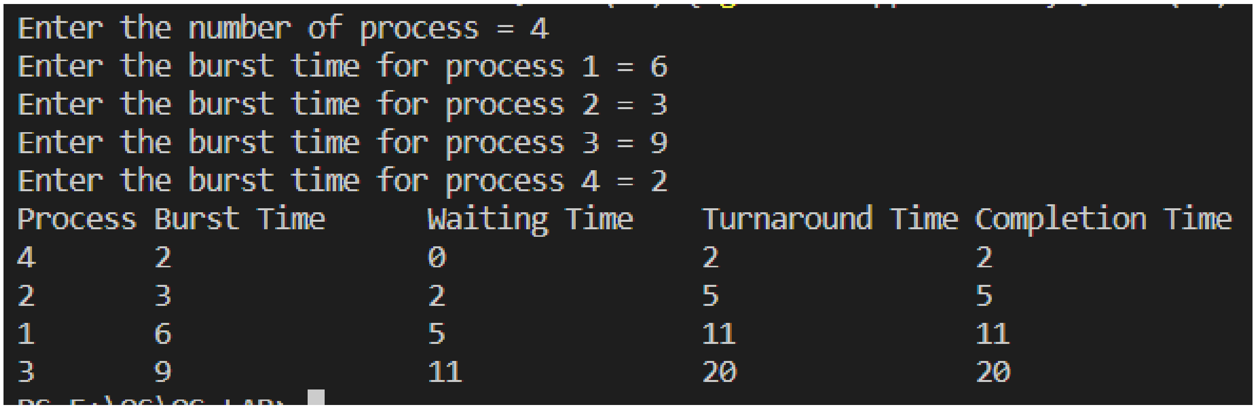
Perumalla Dharan
AP21110010201

1. SJF Implementation



```
#include <iostream>
using namespace std;
struct process
{
    int id;
    int bt;
    int waiting_time=0;
    int turnaround_time=0;
    int completion_time=0;
};
bool compare(process p1, process p2)
{
    return p1.bt < p2.bt;
}
void sjf(process pro[], int n)
{
    for (int i = 0; i < n - 1; i++)
    {
        int min = i;
        for (int j = i + 1; j < n; j++)
        {
            if (pro[j].bt < pro[min].bt)
            {
                min = j;
            }
        }
        swap(pro[min], pro[i]);
    }
    int total_time = 0;
    cout << "Process\tBurst Time\tWaiting Time\tTurnaround Time\tCompletion Time" << endl;
    for (int i = 0; i < n; i++)
    {
        cout << pro[i].id << "\t" << pro[i].bt << "\t\t";
        int waiting_time = total_time;
        cout << waiting_time << "\t\t";
        int turnaround_time = waiting_time + pro[i].bt;
        cout << turnaround_time << "\t\t";
        int completion_time = total_time + pro[i].bt;
        cout << completion_time << endl;
        total_time += pro[i].bt;
    }
}
int main()
{
    int n;
    cout << "Enter the number of process = ";
    cin >> n;
    process pro[n];
    for (int i = 0; i < n; i++)
    {
        cout << "Enter the burst time for process " << i + 1 << " = ";
        cin >> pro[i].bt;
        pro[i].id = i + 1;
        pro[i].waiting_time = 0;
        pro[i].turnaround_time = 0;
        pro[i].completion_time = 0;
    }
    sjf(pro, n);
    return 0;
}
```

Output

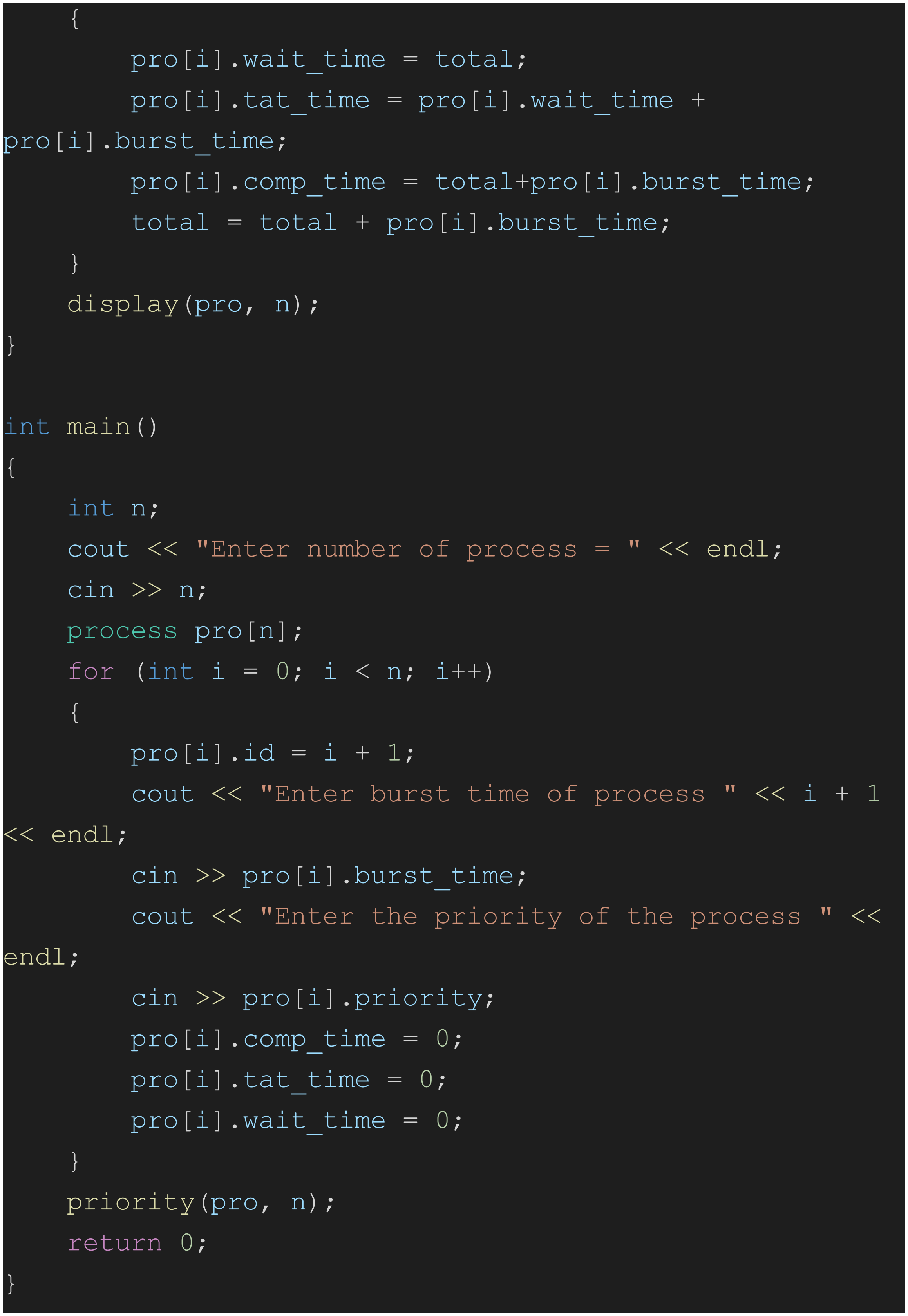


```
Enter the number of process = 4
Enter the burst time for process 1 = 6
Enter the burst time for process 2 = 3
Enter the burst time for process 3 = 9
Enter the burst time for process 4 = 2
Process Burst Time      Waiting Time    Turnaround Time Completion Time
4       2               0               2               2
2       3               2               5               5
1       6               5               11              11
3       9               11              20              20
PS E:\OS\OS LAB>
```

2. Priority Scheduling Implementation

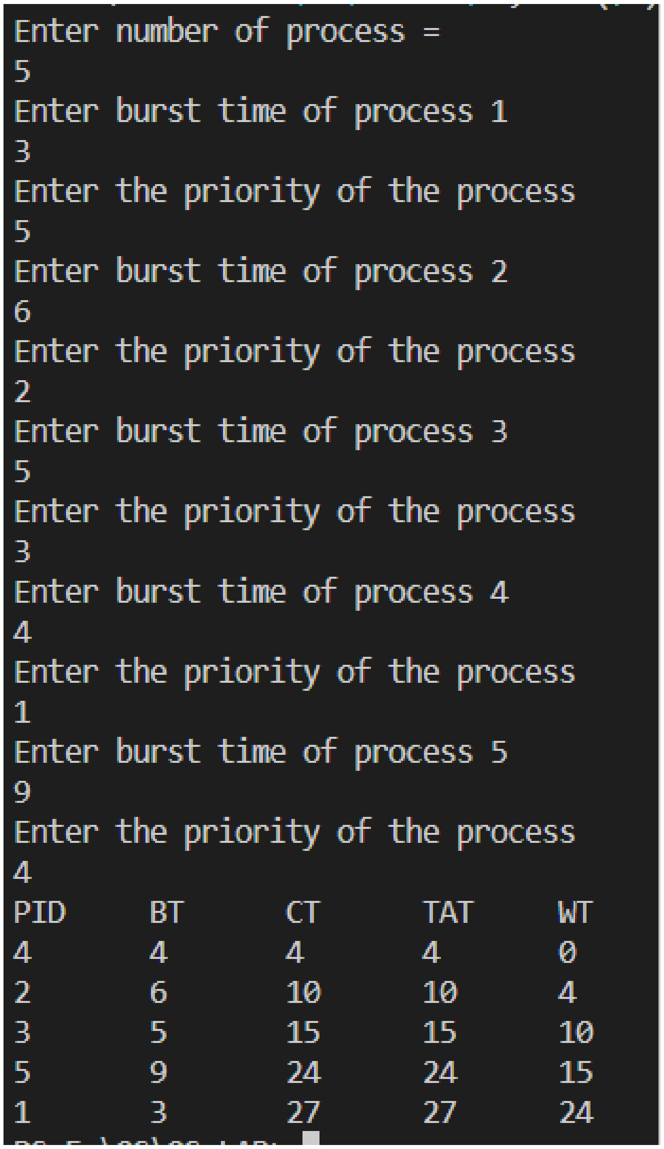
```
#include <iostream>
using namespace std;

struct process
{
    int id, burst_time, wait_time, comp_time, tat_time,
priority;
    process *next;
};

bool compare(process p1, process p2)
```

```
{
    return p1.priority < p2.priority;
}

void display(process p[], int n)
{
    cout << "PID\tBT\tCT\tTAT\tWT" << endl;
    for (int i = 0; i < n; i++)
    {
        cout << p[i].id << "\t" << p[i].burst_time << "\t"
        << p[i].comp_time << "\t" << p[i].tat_time << "\t" << p[i].wait_time << endl;
    }
}

void priority(process pro[], int n)
{
    for (int i = 0; i < n; i++)
    {
        int min = i;
        for (int j = i + 1; j < n; j++)
        {
            if (compare(pro[j], pro[min]))
            {
                min = j;
            }
        }
        swap(pro[min], pro[i]);
    }

    int total = 0;
    for (int i = 0; i < n; i++)
```



```
    {
        pro[i].wait_time = total;
        pro[i].tat_time = pro[i].wait_time +
pro[i].burst_time;
        pro[i].comp_time = total+pro[i].burst_time;
        total = total + pro[i].burst_time;
    }
    display(pro, n);
}

int main()
{
    int n;
    cout << "Enter number of process = " << endl;
    cin >> n;
    process pro[n];
    for (int i = 0; i < n; i++)
    {
        pro[i].id = i + 1;
        cout << "Enter burst time of process " << i + 1
<< endl;
        cin >> pro[i].burst_time;
        cout << "Enter the priority of the process " <<
endl;
        cin >> pro[i].priority;
        pro[i].comp_time = 0;
        pro[i].tat_time = 0;
        pro[i].wait_time = 0;
    }
    priority(pro, n);
    return 0;
}
```

Output



```
Enter number of process =
5
Enter burst time of process 1
3
Enter the priority of the process
5
Enter burst time of process 2
6
Enter the priority of the process
2
Enter burst time of process 3
5
Enter the priority of the process
3
Enter burst time of process 4
4
Enter the priority of the process
1
Enter burst time of process 5
9
Enter the priority of the process
4
PID     BT      CT      TAT     WT
4       4       4       4       0
2       6       10      10      4
3       5       15      15      10
5       9       24      24      15
1       3       27      27      24
```