# Indian Institute of Information Technology Vadodara

**CS266 Operating System Lab**

**Lab Assignment 4**

**Roll No. 201951105** **Name: Nishant Andoriy**

## Problem 1

Implement following CPU scheduling algorithms:

- First Come First Serve
- Shortest Job First
- Priority Based Scheduling

Find the following performance objectives:

- Average Waiting Time
- Average Turnaround Time
- CPU efficiency

**Input:-**

- Number of Processes
- Process Id
- Arrival time of all the processes
  - All arrive at the same time (say 0).
  - All arrive at different time
- Burst time of all the processes

**Output:-**

Table 1: Output Table

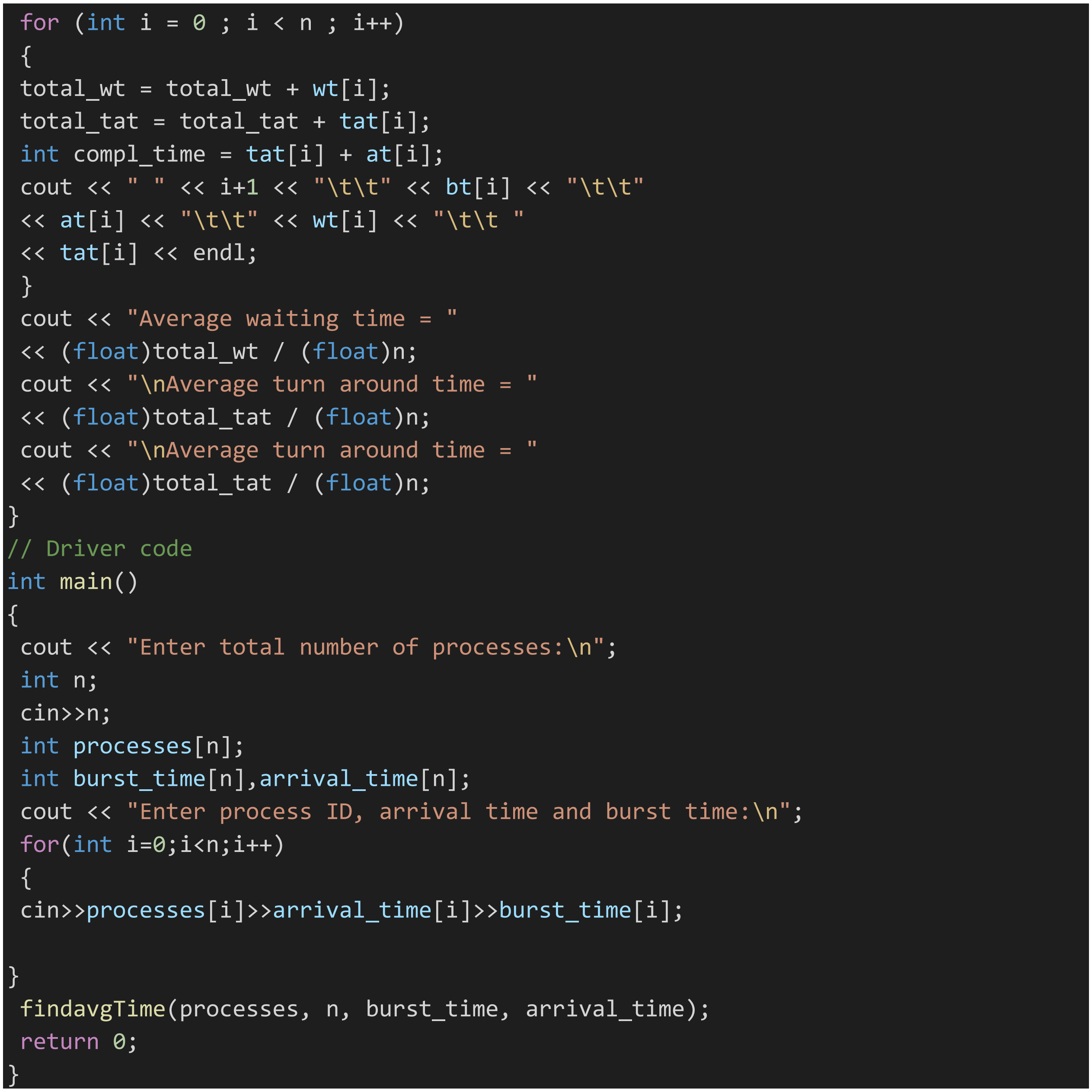
| PID Arrival Time | Burst Time | Waiting Time | Turnaround Time |
|---|---|---|---|
| | | | |

- Average Turnaround Time:-
- Average Waiting Time:-

- **First Come First Serve**

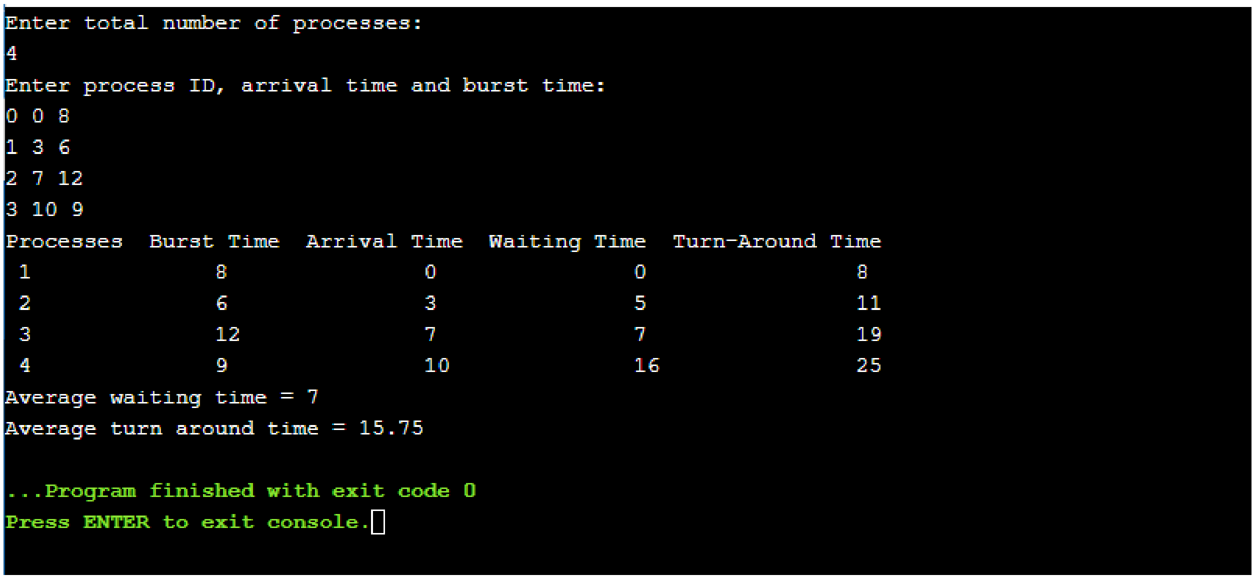
**Ans:-**

```
#include<iostream>
using namespace std;
// Function to find the waiting time for all
// processes
void findWaitingTime(int processes[], int n, int bt[],
 int wt[], int at[])
{
 int service_time[n];
 service_time[0] = 0;
 wt[0] = 0;
 // calculating waiting time
 for (int i = 1; i < n ; i++)
 {
 // Add burst time of previous processes
 service_time[i] = service_time[i-1] + bt[i-1];
 wt[i] = service_time[i] - at[i];
 if (wt[i] < 0)
 wt[i] = 0;
 }
}
// Function to calculate turn around time
void findTurnAroundTime(int processes[], int n, int bt[],
 int wt[], int tat[])
{
 // Calculating turnaround time by adding bt[i] + wt[i]
 for (int i = 0; i < n ; i++)
 tat[i] = bt[i] + wt[i];
}
// Function to calculate average waiting and turn-around
// times.
void findavgTime(int processes[], int n, int bt[], int at[])
{
 int wt[n], tat[n];
 // Function to find waiting time of all processes
 findWaitingTime(processes, n, bt, wt, at);
 // Function to find turn around time for all processes
 findTurnAroundTime(processes, n, bt, wt, tat);
 // Display processes along with all details
 cout << "Processes " << " Burst Time " << " Arrival Time "
 << " Waiting Time " << " Turn-Around Time \n";
 int total_wt = 0, total_tat = 0;
```



```
 for (int i = 0 ; i < n ; i++)
 {
 total_wt = total_wt + wt[i];
 total_tat = total_tat + tat[i];
 int compl_time = tat[i] + at[i];
 cout << " " << i+1 << "\t\t" << bt[i] << "\t\t"
 << at[i] << "\t\t" << wt[i] << "\t\t "
 << tat[i] << endl;
 }
 cout << "Average waiting time = "
 << (float)total_wt / (float)n;
 cout << "\nAverage turn around time = "
 << (float)total_tat / (float)n;
 cout << "\nAverage turn around time = "
 << (float)total_tat / (float)n;
}
// Driver code
int main()
{
 cout << "Enter total number of processes:\n";
 int n;
 cin>>n;
 int processes[n];
 int burst_time[n],arrival_time[n];
 cout << "Enter process ID, arrival time and burst time:\n";
 for(int i=0;i<n;i++)
 {
 cin>>processes[i]>>arrival_time[i]>>burst_time[i];

}
 findavgTime(processes, n, burst_time, arrival_time);
 return 0;
}
```

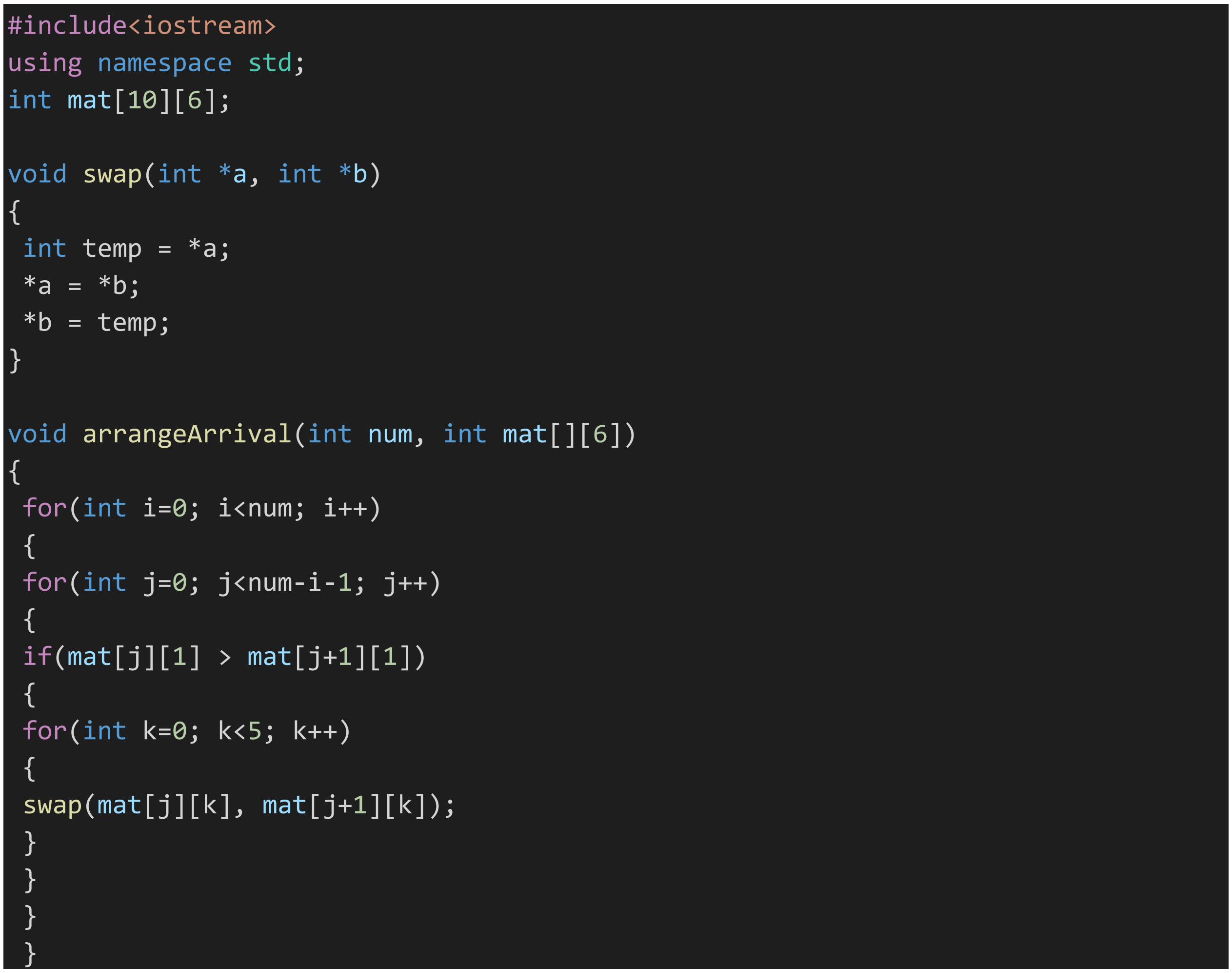
**Output:-**



```
Enter total number of processes:
4
Enter process ID, arrival time and burst time:
0 0 8
1 3 6
2 7 12
3 10 9
Processes  Burst Time  Arrival Time  Waiting Time  Turn-Around Time
 1              8              0              0                 8
 2              6              3              5                 11
 3              12             7              7                 19
 4              9              10             16                25
Average waiting time = 7
Average turn around time = 15.75

...Program finished with exit code 0
Press ENTER to exit console.
```

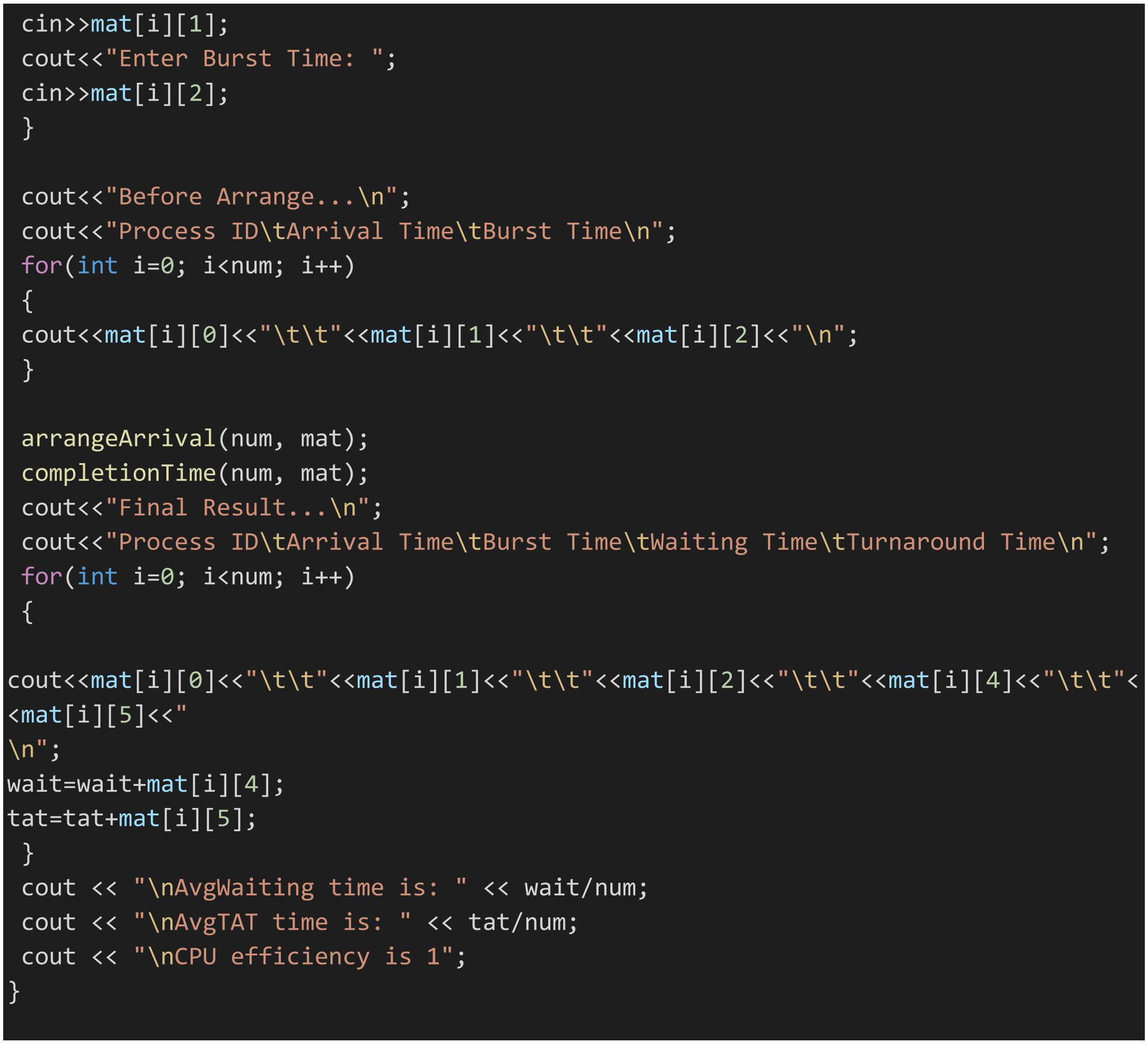
• **Shortest Job First**

**Ans:-**



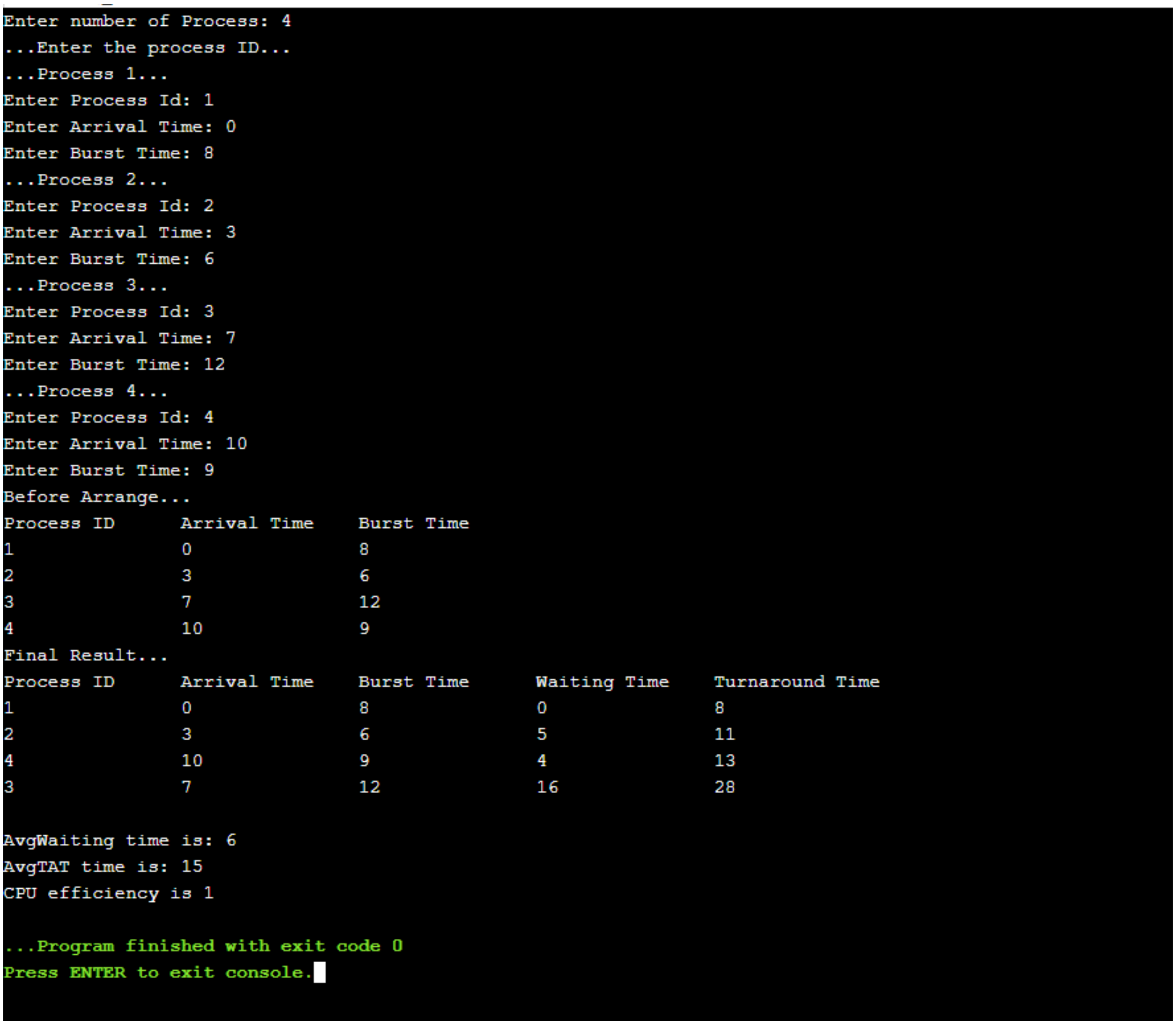
```
#include<iostream>
using namespace std;
int mat[10][6];

void swap(int *a, int *b)
{
 int temp = *a;
 *a = *b;
 *b = temp;
}

void arrangeArrival(int num, int mat[][6])
{
 for(int i=0; i<num; i++)
 {
 for(int j=0; j<num-i-1; j++)
 {
 if(mat[j][1] > mat[j+1][1])
 {
 for(int k=0; k<5; k++)
 {
 swap(mat[j][k], mat[j+1][k]);
 }
 }
 }
 }
```

```
}

void completionTime(int num, int mat[][6])
{
 int temp, val;
 mat[0][3] = mat[0][1] + mat[0][2];
 mat[0][5] = mat[0][3] - mat[0][1];
 mat[0][4] = mat[0][5] - mat[0][2];

 for(int i=1; i<num; i++)
 {
 temp = mat[i-1][3];
 int low = mat[i][2];
 for(int j=i; j<num; j++)
 {
 if(temp >= mat[j][1] && low >= mat[j][2])
 {
 low = mat[j][2];
 val = j;
 }
 }
 mat[val][3] = temp + mat[val][2];
 mat[val][5] = mat[val][3] - mat[val][1];
 mat[val][4] = mat[val][5] - mat[val][2];
 for(int k=0; k<6; k++)
 {
 swap(mat[val][k], mat[i][k]);
 }
 }
}

int main()
{
 int num, temp;
 int wait=0,tat=0;
 cout<<"Enter number of Process: ";
 cin>>num;

 cout<<"...Enter the process ID...\n";
 for(int i=0; i<num; i++)
 {
 cout<<"...Process "<<i+1<<"...\n";
 cout<<"Enter Process Id: ";
 cin>>mat[i][0];
 cout<<"Enter Arrival Time: ";
```



```
 cin>>mat[i][1];
 cout<<"Enter Burst Time: ";
 cin>>mat[i][2];
 }

 cout<<"Before Arrange...\n";
 cout<<"Process ID\tArrival Time\tBurst Time\n";
 for(int i=0; i<num; i++)
 {
 cout<<mat[i][0]<<"\t\t"<<mat[i][1]<<"\t\t"<<mat[i][2]<<"\n";
 }

 arrangeArrival(num, mat);
 completionTime(num, mat);
 cout<<"Final Result...\n";
 cout<<"Process ID\tArrival Time\tBurst Time\tWaiting Time\tTurnaround Time\n";
 for(int i=0; i<num; i++)
 {

cout<<mat[i][0]<<"\t\t"<<mat[i][1]<<"\t\t"<<mat[i][2]<<"\t\t"<<mat[i][4]<<"\t\t"<
<mat[i][5]<<"
\n";
wait=wait+mat[i][4];
tat=tat+mat[i][5];
 }
 cout << "\nAvgWaiting time is: " << wait/num;
 cout << "\nAvgTAT time is: " << tat/num;
 cout << "\nCPU efficiency is 1";
}
```

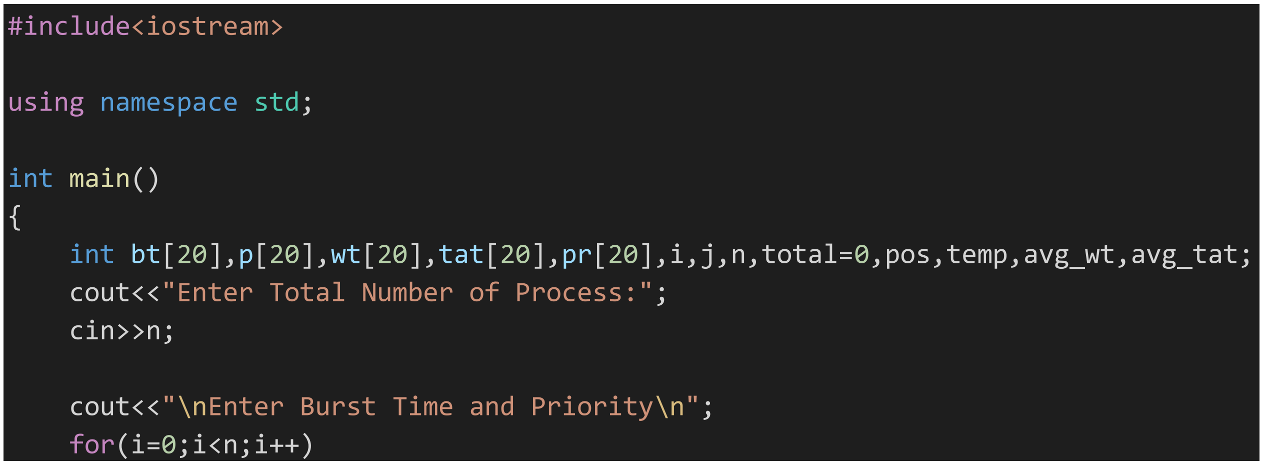
**Output:-**



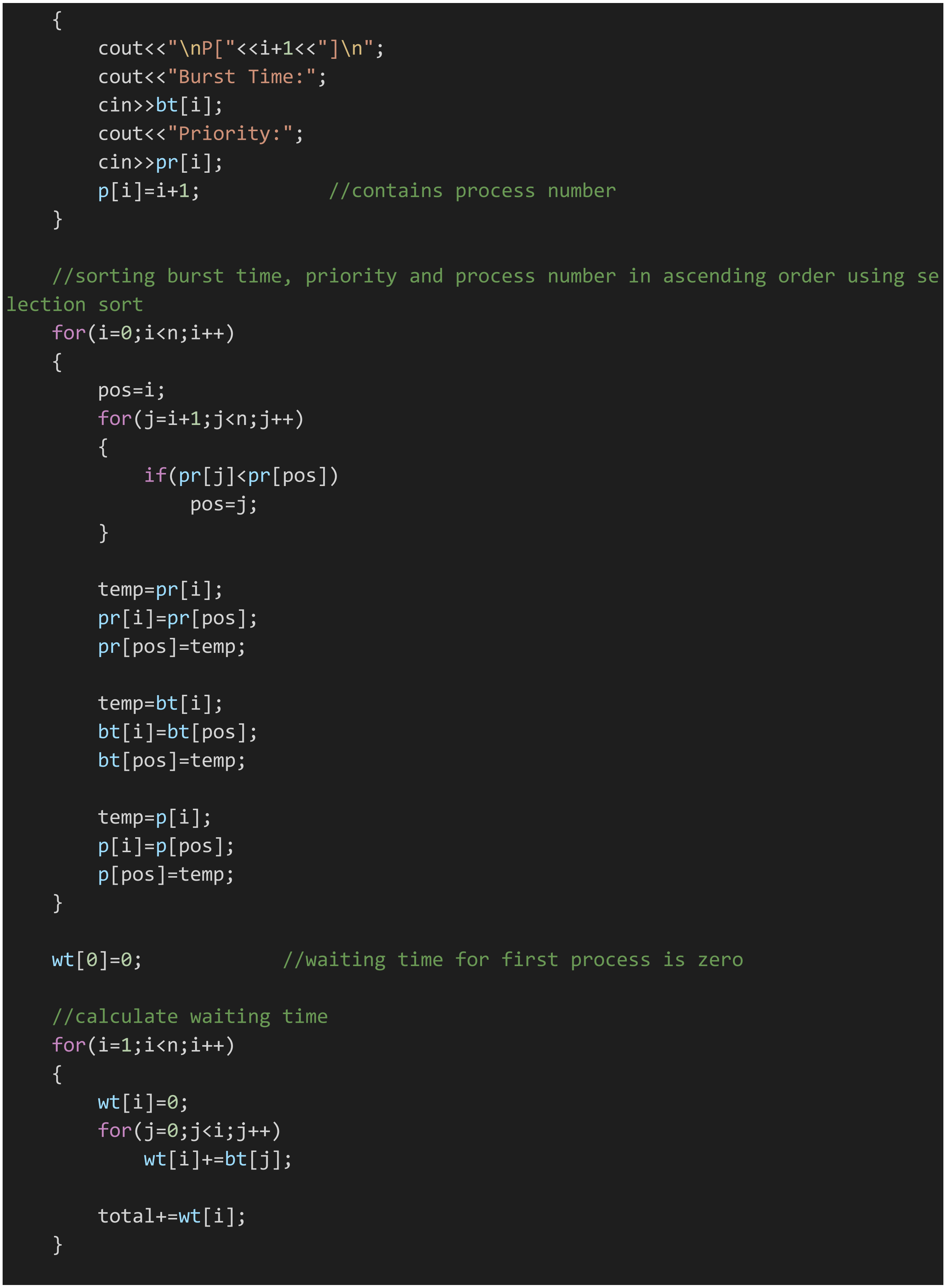
```
Enter number of Process: 4
...Enter the process ID...
...Process 1...
Enter Process Id: 1
Enter Arrival Time: 0
Enter Burst Time: 8
...Process 2...
Enter Process Id: 2
Enter Arrival Time: 3
Enter Burst Time: 6
...Process 3...
Enter Process Id: 3
Enter Arrival Time: 7
Enter Burst Time: 12
...Process 4...
Enter Process Id: 4
Enter Arrival Time: 10
Enter Burst Time: 9
Before Arrange...
Process ID      Arrival Time    Burst Time
1               0               8
2               3               6
3               7               12
4               10              9
Final Result...
Process ID      Arrival Time    Burst Time      Waiting Time    Turnaround Time
1               0               8               0               8
2               3               6               5               11
4               10              9               4               13
3               7               12              16              28

AvgWaiting time is: 6
AvgTAT time is: 15
CPU efficiency is 1

...Program finished with exit code 0
Press ENTER to exit console.
```

• Priority Based Scheduling

Ans:-

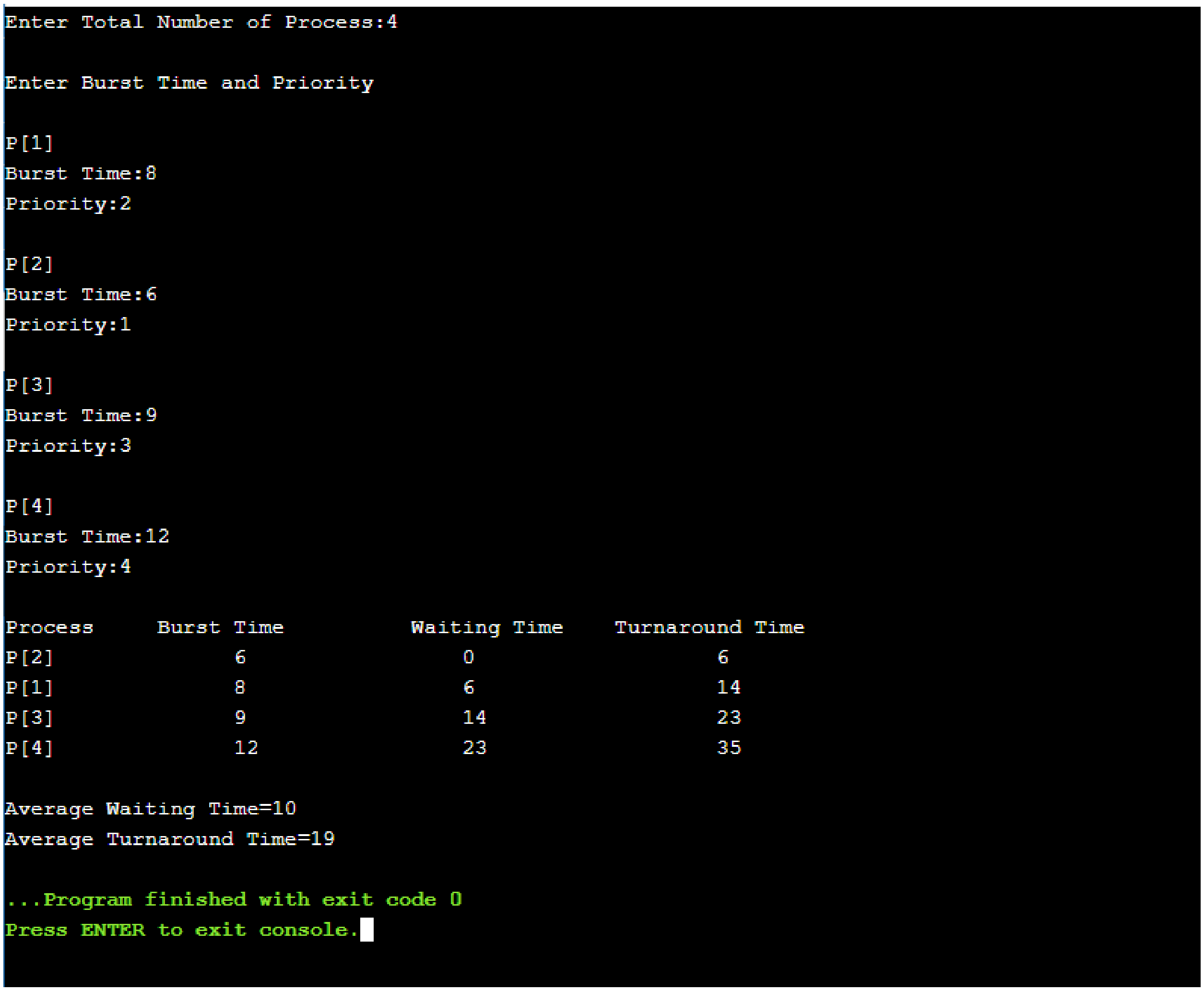


```
#include<iostream>

using namespace std;

int main()
{
    int bt[20],p[20],wt[20],tat[20],pr[20],i,j,n,total=0,pos,temp,avg_wt,avg_tat;
    cout<<"Enter Total Number of Process:";
    cin>>n;

    cout<<"\nEnter Burst Time and Priority\n";
    for(i=0;i<n;i++)
```



```
    {
        cout<<"\nP["<<i+1<<"]\n";
        cout<<"Burst Time:";
        cin>>bt[i];
        cout<<"Priority:";
        cin>>pr[i];
        p[i]=i+1;           //contains process number
    }

    //sorting burst time, priority and process number in ascending order using selection sort
    for(i=0;i<n;i++)
    {
        pos=i;
        for(j=i+1;j<n;j++)
        {
            if(pr[j]<pr[pos])
                pos=j;
        }

        temp=pr[i];
        pr[i]=pr[pos];
        pr[pos]=temp;

        temp=bt[i];
        bt[i]=bt[pos];
        bt[pos]=temp;

        temp=p[i];
        p[i]=p[pos];
        p[pos]=temp;
    }

    wt[0]=0;            //waiting time for first process is zero

    //calculate waiting time
    for(i=1;i<n;i++)
    {
        wt[i]=0;
        for(j=0;j<i;j++)
            wt[i]+=bt[j];

        total+=wt[i];
    }
    avg_wt=total/n;      //average waiting time
    total=0;

    cout<<"\nProcess\t    Burst Time    \tWaiting Time\tTurnaround Time";
    for(i=0;i<n;i++)
    {
        tat[i]=bt[i]+wt[i];     //calculate turnaround time
        total+=tat[i];
        cout<<"\nP["<<p[i]<<"]\t\t  "<<bt[i]<<"\t\t    "<<wt[i]<<"\t\t\t"<<tat[i]
;
    }

    avg_tat=total/n;     //average turnaround time
    cout<<"\n\nAverage Waiting Time="<<avg_wt;
    cout<<"\nAverage Turnaround Time="<<avg_tat;

    return 0;
}
```

**Output:-**



```
Enter Total Number of Process:4

Enter Burst Time and Priority

P[1]
Burst Time:8
Priority:2

P[2]
Burst Time:6
Priority:1

P[3]
Burst Time:9
Priority:3

P[4]
Burst Time:12
Priority:4

Process     Burst Time          Waiting Time    Turnaround Time
P[2]              6                  0                   6
P[1]              8                  6                   14
P[3]              9                  14                  23
P[4]              12                 23                  35

Average Waiting Time=10
Average Turnaround Time=19

...Program finished with exit code 0
Press ENTER to exit console.
```