Name : Ramya . S

Department : Bsc Information Technology

Roll No : 18bit143

Sri Krishna arts and science college

**GIVEN PROBLEM STATEMENT :**

Implement a list scheduling algorithm , assign priority to the process and simulate the working using C++ from scratch. Share the code and time complexity achieved with a neat documentation.

**CODE :**

#include<iostream>

using namespace std;

int main()

{

int bt[20],p[20],wt[20],tat[20],pr[20],i,j,n,sum=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;

}

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;

for(i=1;i<n;i++)

{

wt[i]=0;

for(j=0;j<i;j++)

wt[i]+=bt[j];

sum+=wt[i];

}

avg\_wt=sum/n;

sum=0;

cout<<"\nProcess\t Burst Time \tWaiting Time\tTurnaround Time";

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

{

tat[i]=bt[i]+wt[i];

sum+=tat[i];

cout<<"\nP["<<p[i]<<"]\t\t "<<bt[i]<<"\t\t "<<wt[i]<<"\t\t\t"<<tat[i];

}

avg\_tat=sum/n;

cout<<"\n\nAverage Waiting Time="<<avg\_wt;

cout<<"\nAverage Turnaround Time="<<avg\_tat;

cout<<"\n The order of processes based on the scheduled priority";

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

{

cout<<"\nP["<<p[i]<<"]\n";

}

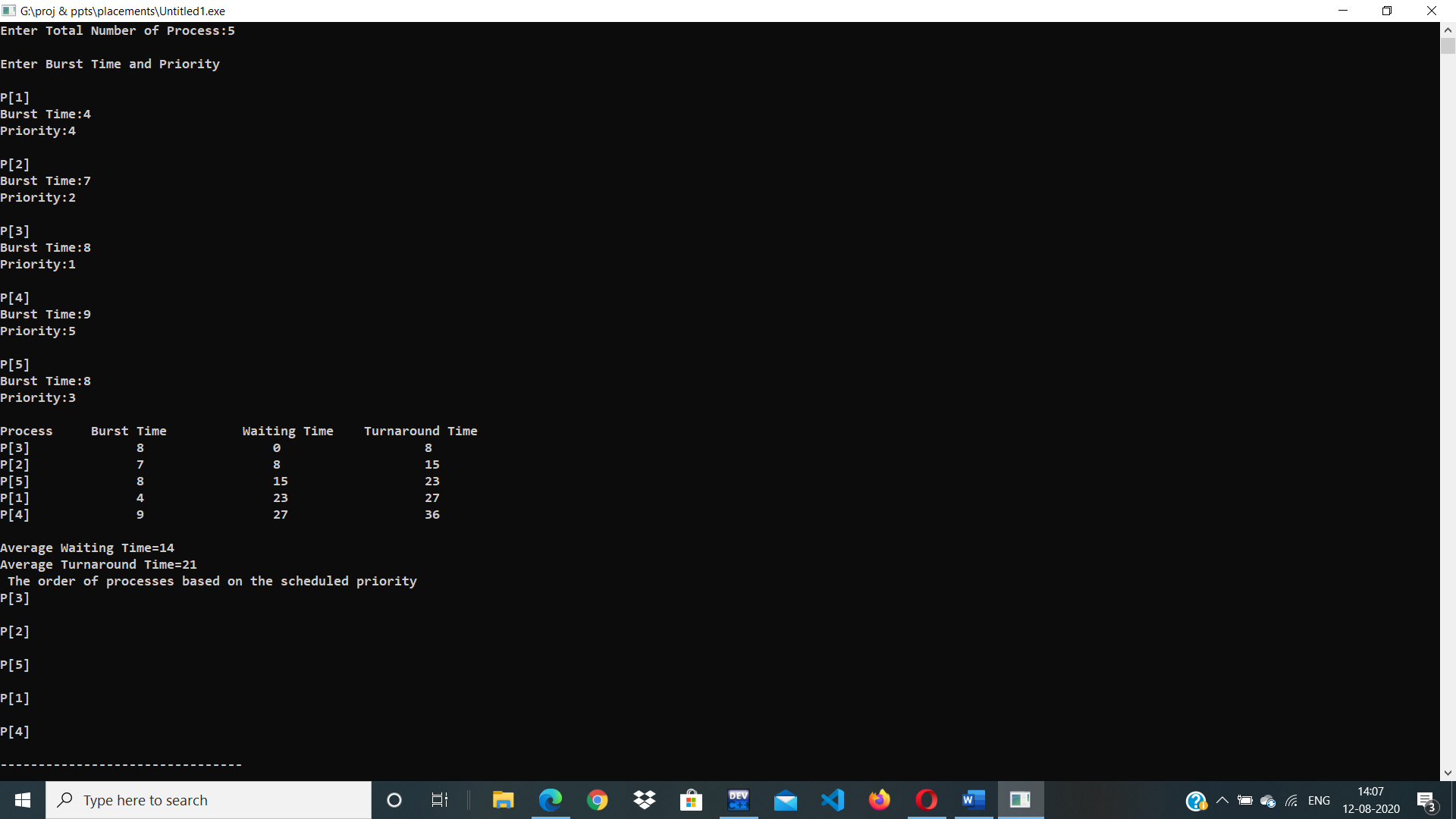
return 0;

}

**EXPLANATION :**

* This code demonstrates list scheduling algorithm by assigning priorities to the given processes. The CPU schedules the process based on the order of priorities assigned to each process. The highest priority process is executed first.
* Total number of process, burst time and priority of each process is obtained as input. The burst time is sorted in ascending by using “selection sort” based on the priority .Waiting time and turnaround time is calculated for each of the given process.
* Finally the average wait time , average turnaround time and the list of processes based on its priority are displayed.

**OUTPUT :**



**TIME COMPLEXITY :**

The time complexity of this program is O(n^2).