**First Come First Serve Scheduling**

**Program:**

#include<iostream>

using namespace std;

int main()

{

int n,bt[20],wt[20],tat[20],avwt=0,avtat=0,i,j;

cout<<"Enter total number of processes(maximum 20):";

cin>>n;

cout<<"Enter Process Burst Time";

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

{

cout<<"P["<<i+1<<"]:";

cin>>bt[i];

}

wt[0]=0;

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

{

wt[i]=0;

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

wt[i]+=bt[j];

}

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

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

{

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

avwt+=wt[i];

avtat+=tat[i];

cout<<"\nP["<<i+1<<"]"<<"\t\t"<<bt[i]<<"\t\t"<<wt[i]<<"\t\t"<<tat[i];

}

avwt/=i;

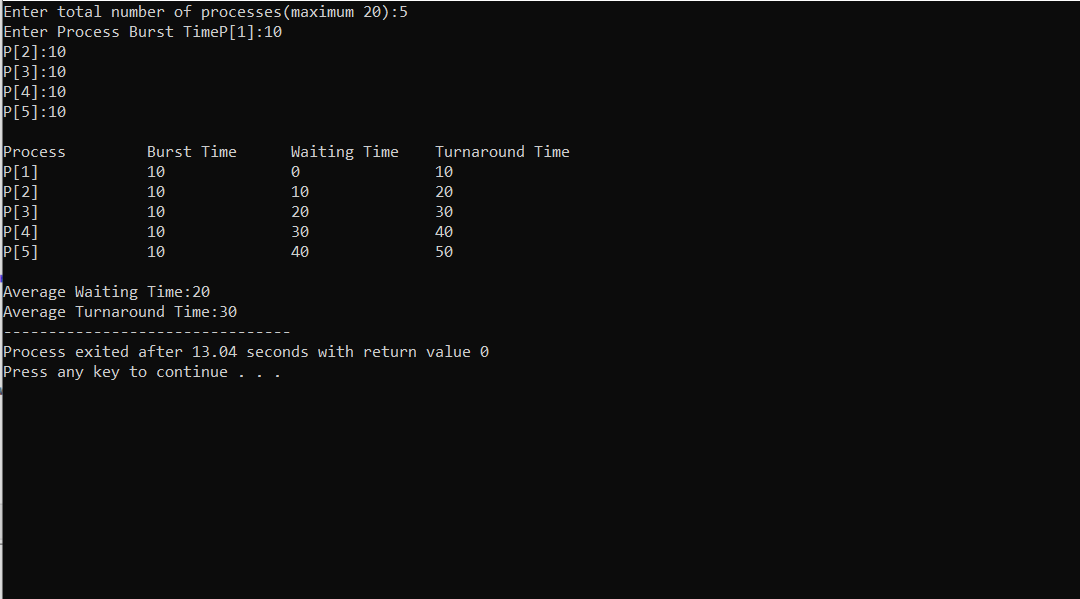
avtat/=i;

cout<<"\n\nAverage Waiting Time:"<<avwt;

cout<<"\nAverage Turnaround Time:"<<avtat;

return 0;

}

**Output:**

**Shortest Job First Scheduling**

**Program:**

#include<iostream>

using namespace std;

int main()

{

int n,temp,tt=0,min,d,i,j;

float atat=0,awt=0,stat=0,swt=0;

cout<<"enter no of process"<<endl;

cin>>n;

int a[n],b[n],e[n],tat[n],wt[n];

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

{

cout<<"enter arival time ";

cin>>a[i];

}

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

{

cout<<"enter brust time ";

cin>>b[i];

}

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

{

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

{

if(b[i]>b[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

temp=b[i];

b[i]=b[j];

b[j]=temp;

}

}

}

min=a[0];

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

{

if(min>a[i])

{

min=a[i];

d=i;

}

}

tt=min;

e[d]=tt+b[d];

tt=e[d];

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

{

if(a[i]!=min)

{

e[i]=b[i]+tt;

tt=e[i];

}

}

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

{

tat[i]=e[i]-a[i];

stat=stat+tat[i];

wt[i]=tat[i]-b[i];

swt=swt+wt[i];

}

atat=stat/n;

awt=swt/n;

cout<<"Process Arrival-time(s) Burst-time(s) Waiting-time(s) Turnaround-time(s)\n";

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

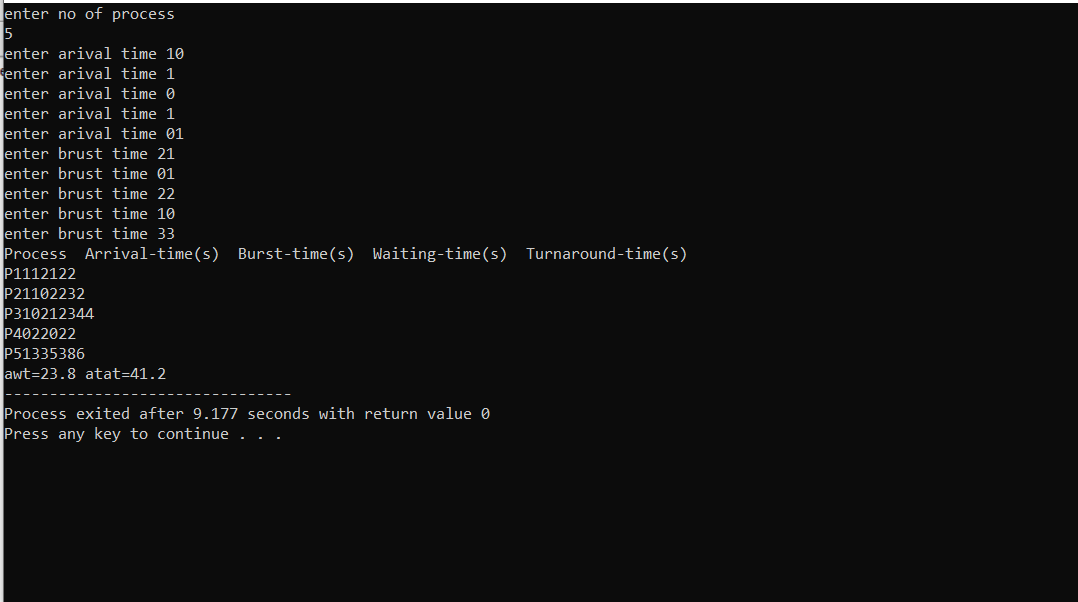
{

cout<<"P"<<i+1<<""<<a[i]<<""<<b[i]<<""<<wt[i]<<""<<tat[i]<<endl;

}

cout<<"awt="<<awt<<" atat="<<atat;

}

****

**Round Robin Scheduling**

**Program:**

#include<stdio.h>

int main()

{

int count,j,n,time,remain,flag=0,time\_quantum;

int wait\_time=0,turnaround\_time=0,at[10],bt[10],rt[10];

printf("Enter Total Process:\t ");

scanf("%d",&n);

remain=n;

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

{

printf("Enter Arrival Time and Burst Time for Process Process Number %d :",count+1);

scanf("%d",&at[count]);

scanf("%d",&bt[count]);

rt[count]=bt[count];

}

printf("Enter Time Quantum:\t");

scanf("%d",&time\_quantum);

printf("\n\nProcess\t|Turnaround Time|Waiting Time\n\n");

for(time=0,count=0;remain!=0;)

{

if(rt[count]<=time\_quantum && rt[count]>0)

{

time+=rt[count];

rt[count]=0;

flag=1;

}

else if(rt[count]>0)

{

rt[count]-=time\_quantum;

time+=time\_quantum;

}

if(rt[count]==0 && flag==1)

{

remain--;

printf("P[%d]\t|\t%d\t|\t%d\n",count+1,time-at[count],time-at[count]-bt[count]);

wait\_time+=time-at[count]-bt[count];

turnaround\_time+=time-at[count];

flag=0;

}

if(count==n-1)

count=0;

else if(at[count+1]<=time)

count++;

else

count=0;

}

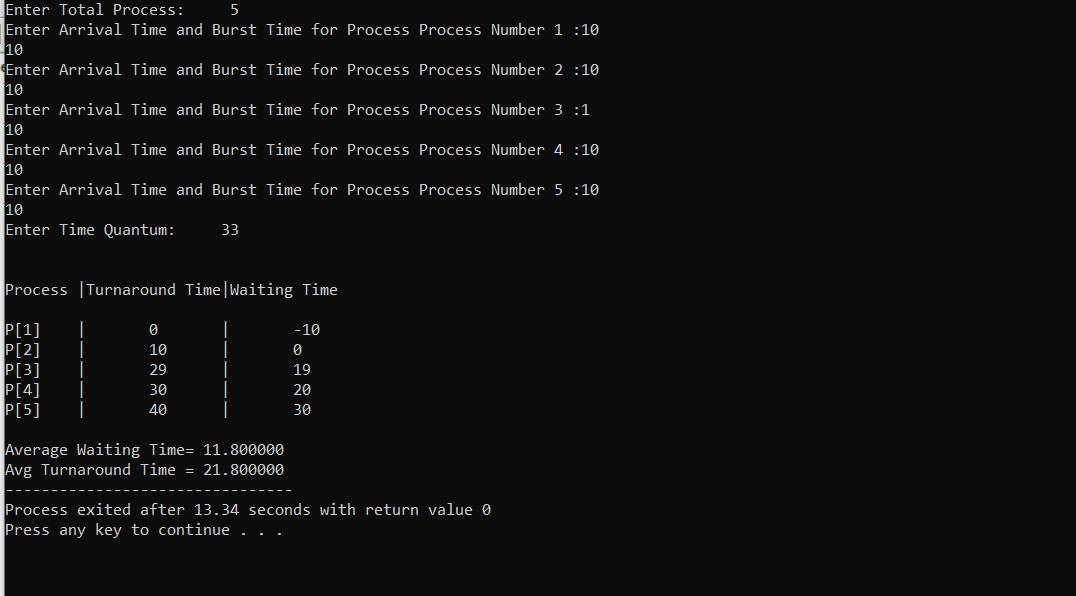
printf("\nAverage Waiting Time= %f\n",wait\_time\*1.0/n);

printf("Avg Turnaround Time = %f",turnaround\_time\*1.0/n);

return 0;

}

**Output:**

****