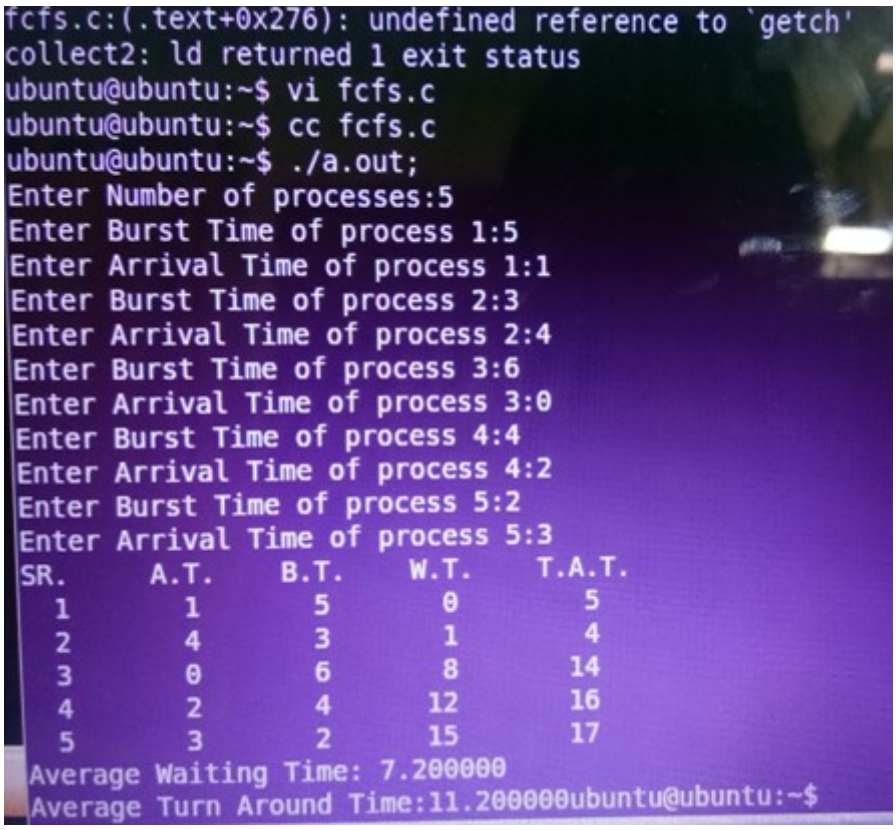
EXPERIMENT: 4

NAME: KRISHNA KUMAR MAHTO
REG. NO. :- 16BIT0453

First come first serve algorithm

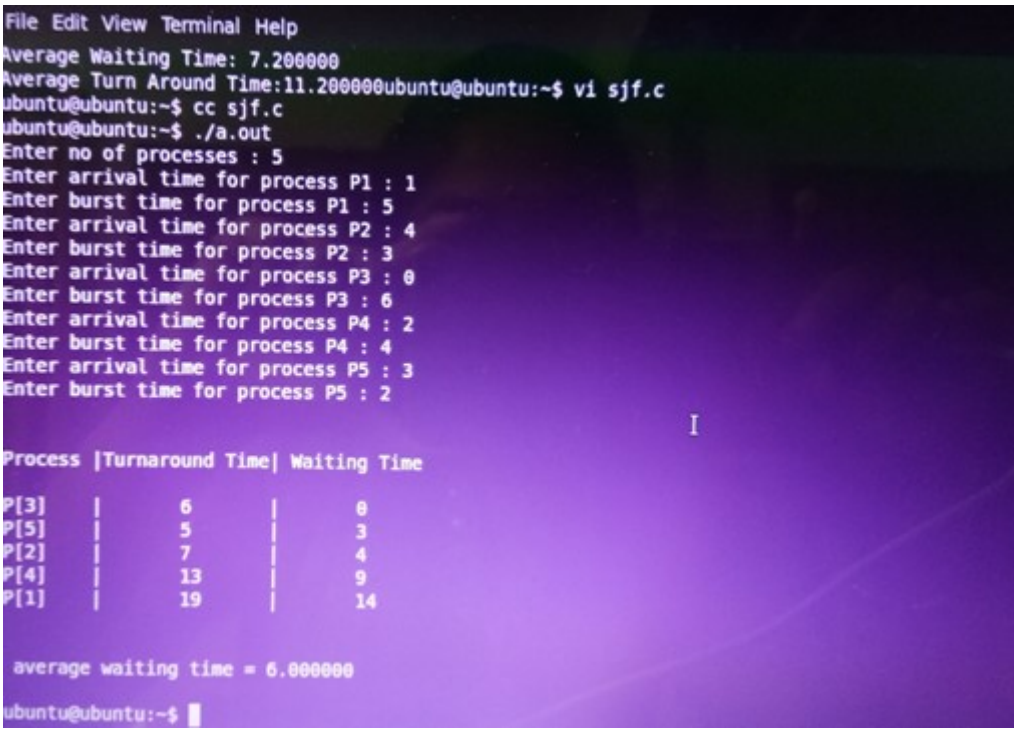
```
#include <stdio.h>
#include <conio.h>
intwt[10],bt[10],at[10],tat[10],n;
float awt,atat;
void input(){
printf("Enter Number of processes:");
scanf("%d",&n);
inti;
for(i=0;i<n;i++)
{
printf("Enter Burst Time of process %d:",i+1);
scanf("%d",&bt[i]);
printf("Enter Arrival Time of process %d:",i+1);
scanf("%d",&at[i]);
}
}
void calculate(){
wt[0]=0;
atat=tat[0]=bt[0];
intbtt=bt[0];
inti;
for(i=1;i<n;i++){wt[i]=btt-at[i];
btt+=bt[i];
awt+=wt[i];
tat[i]= wt[i]+bt[i];
atat+=tat[i];
}
atat/=n;
awt/=n;
}
void display(){
inti;
printf("SR.\tA.T.\tB.T.\tW.T.\tT.A.T.\n");
for(i=0;i<n;i++)
{
printf("%3d\t%3d\t%3d\t%3d\t%4d\n",i+1,at[i],bt[i],wt[i],tat[i]);
}
printf("Average Waiting Time: %f\nAverage Turn Around Time:%f",awt,atat);
}
intmain(){
input();
calculate();
display();
getch();
}
```



Shortest job first algorithm non-preemptive

```
#include<stdio.h>
intmain()
{
inttime,bt[10],at[10],sum_bt=0,smallest,n,i;
intsum_turnaround=0,sum_wait=0;
printf("Enter no of processes : ");
scanf("%d",&n);for(i=0;i<n;i++)
{
printf("Enter arrival time for process P%d : ",i+1);
scanf("%d",&at[i]);
printf("Enter burst time for process P%d : ",i+1);
scanf("%d",&bt[i]);
sum_bt+=bt[i];
}
bt[9]=9999;
printf("\n\nProcess\t|Turnaround Time| Waiting Time\n\n");
for(time=0;time<sum_bt;)
{
smallest=9;
for(i=0;i<n;i++)
{
if(at[i]<=time &&bt[i]>0 &&bt[i]<bt[smallest])
smallest=i;
}
if(smallest==9)
```

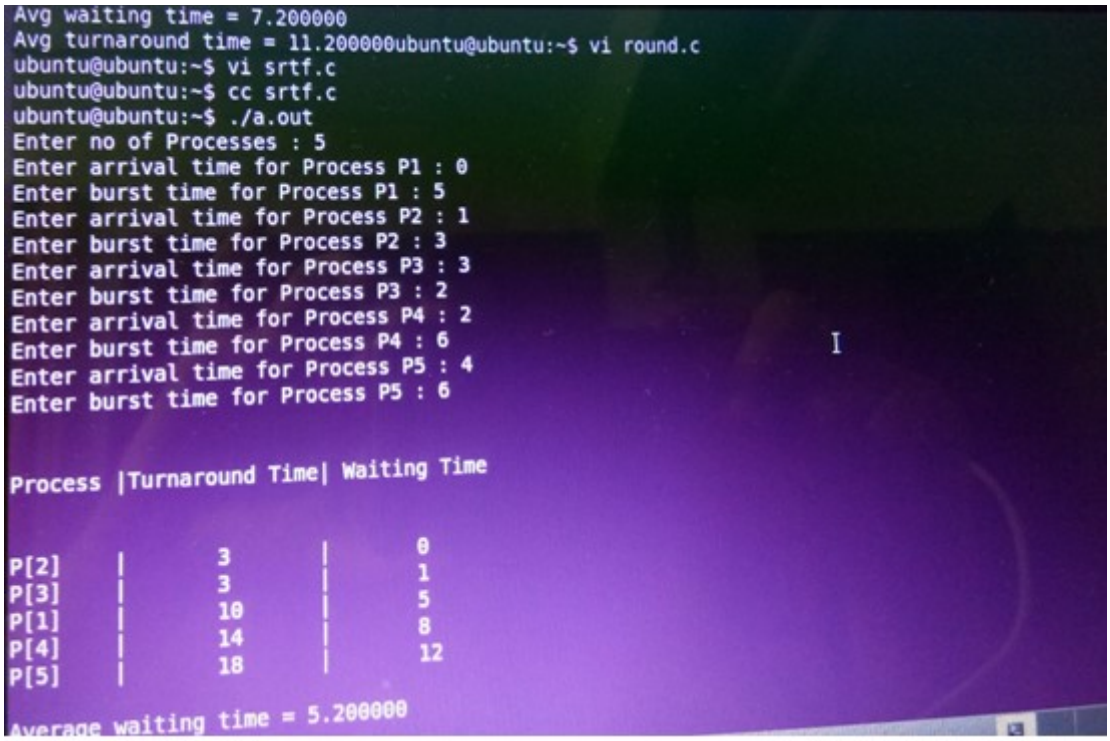
```
{
time++;
continue;
}
printf("P[%d]\t|\t%d\t|\t%d\n",smallest+1,time+bt[smallest]-at[smallest],time-at[smallest]);
sum_turnaround+=time+bt[smallest]-at[smallest];
sum_wait+=time-at[smallest];
time+=bt[smallest];
bt[smallest]=0;
}
printf("\n\n average waiting time = %f",sum_wait*1.0/n);printf("\n\n average turnaround time =
%f",sum_turnaround*1.0/n);
return 0;
}
```



Shortest remaining time first algorithm(preemptive)

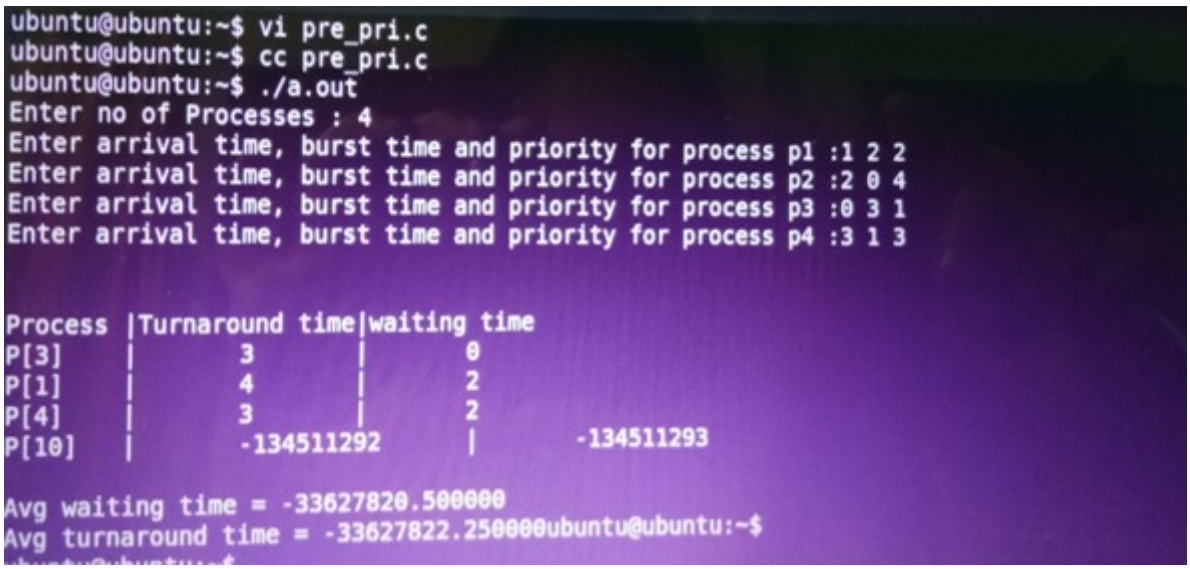
```
#include<stdio.h>
int main()
{
int at[10],bt[10],rt[10],endTime,i,smallest;
int remain=0,n,time,sum_wait=0,sum_turnaround=0;
printf("Enter no of Processes : ");
scanf("%d",&n);
for(i=0;i<n;i++)
{printf("Enter arrival time for Process P%d : ",i+1);
scanf("%d",&at[i]);
printf("Enter burst time for Process P%d : ",i+1);
scanf("%d",&bt[i]);
rt[i]=bt[i];
}
```

```
printf("\n\nProcess\t|Turnaround Time| Waiting Time\n\n");
rt[9]=9999;
for(time=0;remain!=n;time++)
{
smallest=9;
for(i=0;i<n;i++)
{
if(at[i]<=time && rt[i]<rt[smallest] && rt[i]>0)
{
smallest=i;
}
}
rt[smallest]--;
if(rt[smallest]==0)
{
remain++;
endTime=time+1;
printf("\nP[%d]\t|\t%d\t|\t%d",smallest+1,endTime-at[smallest],endTime-bt[smallest]-
at[smallest]);
sum_wait+=endTime-bt[smallest]-at[smallest];
sum_turnaround+=endTime-at[smallest];
}
}
printf("\n\nAverage waiting time = %f\n",sum_wait*1.0/n);
printf("Average Turnaround time = %f",sum_turnaround*1.0/5);return 0;
}
```



Preemptive priority algorithm

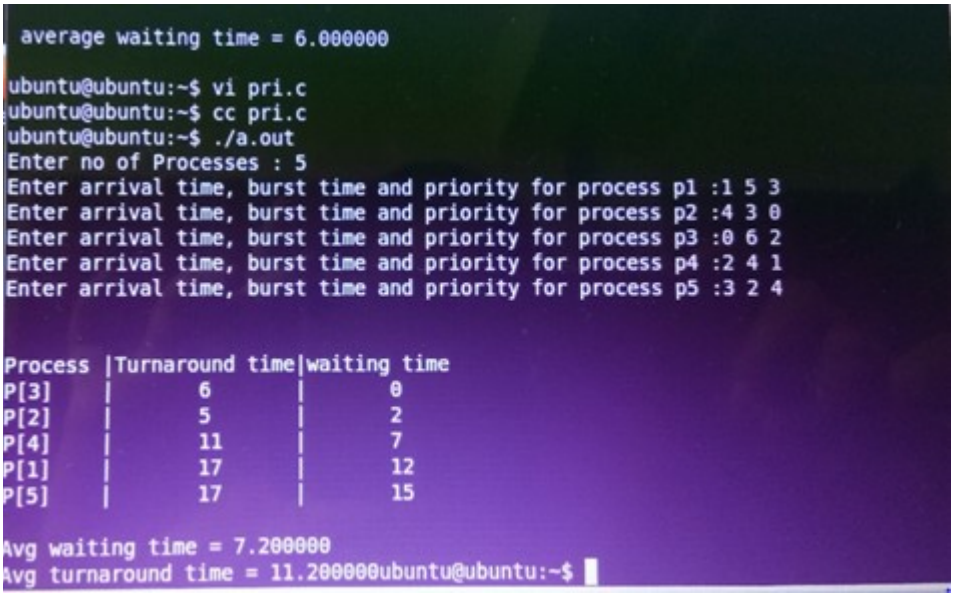
```
#include<stdio.h>
intmain()
{inti,j,n,time,sum_wait=0,sum_turnaround=0,smallest;
int at[10],bt[10],pt[10],rt[10],remain;
printf("Enter no of Processes : ");
scanf("%d",&n);
remain=n;
for(i=0;i<n;i++)
{
printf("Enter arrival time, burst time and priority for process p%d :",i+1);
scanf("%d",&at[i]);
scanf("%d",&bt[i]);
scanf("%d",&pt[i]);
rt[i]=bt[i];
}
pt[9]=11;
printf("\n\nProcess\t|Turnaroundtime|waiting time\n");
for(time=0;remain!=0;time++)
{
smallest=9;
for(i=0;i<n;i++)
{
if(at[i]<=time &&pt[i]<pt[smallest] &&rt[i]>0)
{
smallest=i;
}
}
rt[smallest]--;
if(rt[smallest]==0)
{
remain--;
printf("P[%d]\t|\t%d\t|\t%d\n",smallest+1,time+1-at[smallest],time+1-at[smallest]-
bt[smallest]);sum_wait+=time+1-at[smallest];
sum_turnaround+=time+1-at[smallest]-bt[smallest];
}
}
printf("\nAvg waiting time = %f\n",sum_wait*1.0/n);
printf("Avg turnaround time = %f",sum_turnaround*1.0/n);
return 0;
}
```



Non-preemptive priority algorithm

```
#include<stdio.h>
intmain()
{
inti,j,n,time,sum_wait=0,sum_turnaround=0;
intsmallest,at[10],bt[10],priority[10],remain;
printf("Enter no of Processes : ");
scanf("%d",&n);
remain=n;for(i=0;i<n;i++)
{
printf("Enter arrival time, burst time and priority for process p%d :",i+1);
scanf("%d",&at[i]);
scanf("%d",&bt[i]);
scanf("%d",&priority[i]);
}
priority[9]=11;
printf("\n\nProcess\t|Turnaroundtime|waiting time\n");
for(time=0;remain!=0;)
{
smallest=9;
for(i=0;i<n;i++)
{
if(at[i]<=time && priority[i]<priority[smallest] &&bt[i]>0)
{
smallest=i;
}
}
time+=bt[smallest];
remain--;
printf("P[%d]\t|\t%d\t|\t%d\n",smallest+1,time-at[smallest],time-at[smallest]-bt[smallest]);
sum_wait+=time-at[smallest]-bt[smallest];
sum_turnaround+=time-at[smallest];
bt[smallest]=0;
}
```

```
printf("\nAvg waiting time = %f\n",sum_wait*1.0/n);
printf("Avg turnaround time = %f",sum_turnaround*1.0/n);
return 0;
}
```



Round robin algorithm

```
#include<stdio.h>
int main()
{
intcount,j,n,time,remain,flag=0,time_quantum;
intwait_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|TurnaroundTime|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;
}
```