

|  |  |  |
| --- | --- | --- |
| **NAME :**  Sriram Vaishnav R | **REG NO :**  21BRS1378 | **MAIL ID :**  sriramvaishnav.r2021@vitstudent.ac.in |
| **PROGRAMME :**  BTech-CSE with spl. in Artificial Intelligence and Robotics | **SCHOOL :**  School of Computer Science and Engineering(SCOPE) | **SEM :**  WS (’22 – ’23) |
| **FACULTY :**  Dr Anandan P | **COURSE :**  BCSE303P | **SLOT :**  (L43 + L44) |

**Assessment-3**

**Scheduling**

**Questions:**

**Get the arrival Time and the Burst Time of 5 processes and execute the following scheduling algorithms:**

**FCFS:**

**Code:**

#include <stdio.h>

#include <conio.h>

int main()

{

int n=5,bt[20],at[20],wt[20],tat[20]; int i,j,total=0,pos,temp;

float avg\_wt,avg\_tat;

printf("Enter Arrival Time and Burst Time for 5 Processes:\n"); for(i=0;i<n;i++)

{

printf("\nP[%d]\n",i+1); printf("Arrival Time: "); scanf("%d",&at[i]); printf("Burst Time: "); scanf("%d",&bt[i]);

}

pos=i;

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

{

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

{

if(at[j]<at[pos])

pos=j;

}

temp=at[i]; at[i]=at[pos]; at[pos]=temp;

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

} wt[0]=0;

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

{

wt[i]=0; for(j=0;j<i;j++)

wt[i]+=bt[j]; total+=wt[i];

}

avg\_wt=(float)total/n; total=0;

printf("\nProcess\t Arrival Time \t Burst Time \t Waiting Time \t Turnaround Time"); for(i=0;i<n;i++)

{

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

printf("\nP[%d]\t\t %d\t\t %d\t\t %d\t\t\t %d",i+1,at[i],bt[i],wt[i],tat[i]);

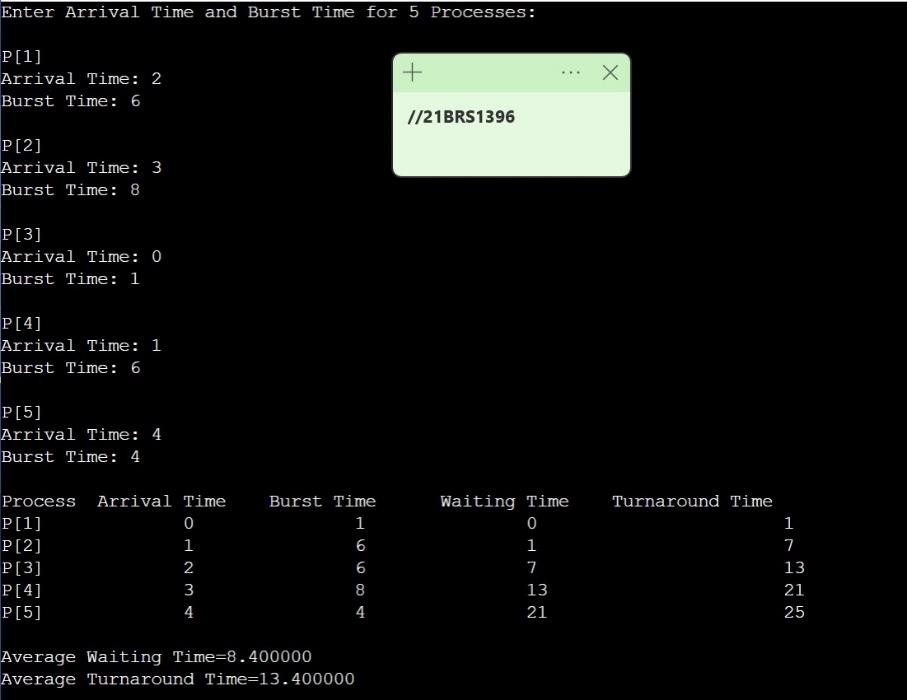
}

avg\_tat=(float)total/n; printf("\n\nAverage Waiting Time=%f",avg\_wt);

printf("\nAverage Turnaround Time=%f\n",avg\_tat); return 0;

}

**Output of FCFS:**



21BRS1378

**Non-Preemptive SJF:**

**Code:**

#include<stdio.h>

int main() {

int time, burst\_time[10], at[10], sum\_burst\_time = 0, smallest, n, i; int sumt = 0, sumw = 0;

printf("enter the no of processes : "); scanf("%d", & n);

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

printf("the arrival time for process P%d : ", i + 1); scanf("%d", & at[i]);

printf("the burst time for process P%d : ", i + 1); scanf("%d", & burst\_time[i]);

sum\_burst\_time += burst\_time[i];

}

burst\_time[9] = 9999;

printf("Process\t|\tArrival Time\t|\tBurst Time\t|\tTurnaround Time\t|\tWaiting Time\n");

for (time = 0; time < sum\_burst\_time;) { smallest = 9;

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

if (at[i] <= time && burst\_time[i] > 0 && burst\_time[i] < burst\_time[smallest]) smallest = i;

}

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

sumt += time + burst\_time[smallest] - at[smallest];

sumw += time - at[smallest]; time += burst\_time[smallest]; burst\_time[smallest] = 0;

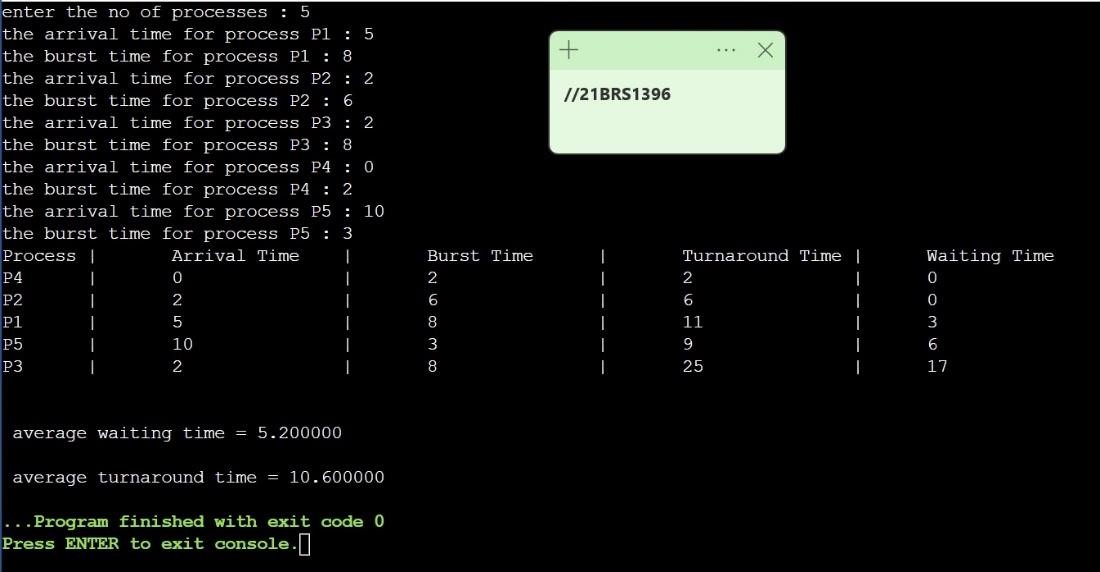
}

printf("\n\n average waiting time = %f", sumw \* 1.0 / n); printf("\n\n average turnaround time = %f", sumt \* 1.0 / n); return 0;

}

**Output for Non-Preemptive SJF:**

21BRS1378



**Preemptive SJF:**

**Code:**

#include <stdio.h> int main()

{

int arrival\_time[10], burst\_time[10], temp[10],bt[10],tt,wt; int i, smallest, count = 0, time, n;

double wait\_time = 0, turnaround\_time = 0, end;

float average\_waiting\_time, average\_turnaround\_time;

printf("\nEnter No. of Processes:\t"); scanf("%d", &n);

printf("\nEnter Details of %d Processes", n); for(i = 0; i < n; i++)

{

printf("\nEnter Arrival Time:\t"); scanf("%d", &arrival\_time[i]); printf("Enter Burst Time:\t"); scanf("%d", &burst\_time[i]); temp[i] = burst\_time[i];

}

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

{

bt[i]=burst\_time[i];

}

burst\_time[9] = 9999; printf("JOB\t|\tAT\t|\tBT\t|\tTAT\t|\tWT\t\n"); for(time = 0; count != n; time++)

{

smallest = 9;

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

{

if(arrival\_time[i] <= time && burst\_time[i] < burst\_time[smallest] && burst\_time[i] > 0)

{

smallest = i;

}

}

burst\_time[smallest]--; if(burst\_time[smallest] == 0)

{

count++;

end = time + 1;

wait\_time = wait\_time + end - arrival\_time[smallest] - temp[smallest]; turnaround\_time = turnaround\_time + end - arrival\_time[smallest]; wt=end - arrival\_time[smallest] - temp[smallest];

tt=end - arrival\_time[smallest];

printf("P%d\t|\t%d\t|\t%d\t|\t%d\t|\t%d\n",smallest + 1,arrival\_time[smallest],bt[smallest],tt, wt);

}

}

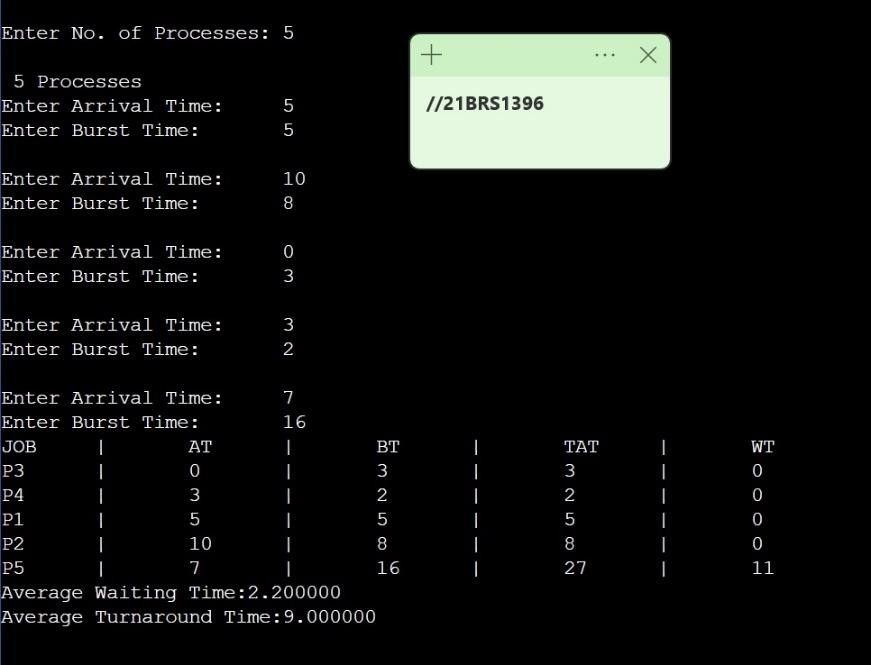
average\_waiting\_time = wait\_time / n; average\_turnaround\_time = turnaround\_time / n; printf("Average Waiting Time:%f\n", average\_waiting\_time);

printf("Average Turnaround Time:%f\n", average\_turnaround\_time); return 0;

}

**Output for Preemptive SJF:**

21BRS1378



**Non-Preemptive Priority:**

**Code:**

#include <stdio.h> #include <stdio.h> void swap(int \*a,int \*b)

{

int temp=\*a;

\*a=\*b;

\*b=temp;

}

int main()

{

int n;

printf("Enter Number of Processes: "); scanf("%d",&n);

int b[n],p[n],index[n]; for(int i=0;i<n;i++)

{

printf("Enter Burst Time and Priority Value for Process %d: ",i+1); scanf("%d %d",&b[i],&p[i]);

index[i]=i+1;

}

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

{

int a=p[i],m=i; for(int j=i;j<n;j++)

{

if(p[j] > a)

{

a=p[j]; m=j;

}

}

swap(&p[i], &p[m]);

swap(&b[i], &b[m]);

swap(&index[i],&index[m]);

}

int t=0;

printf("Order of process Execution is\n"); for(int i=0;i<n;i++)

{

printf("P%d is executed from %d to %d\n",index[i],t,t+b[i]); t+=b[i];

}

printf("\n");

printf("Process Id Burst Time Wait Time TurnAround Time\n"); int wait\_time=0;

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

{

printf("P%d\t\t%d\t\t%d\t\t%d\n",index[i],b[i],wait\_time,wait\_time + b[i]); wait\_time += b[i];

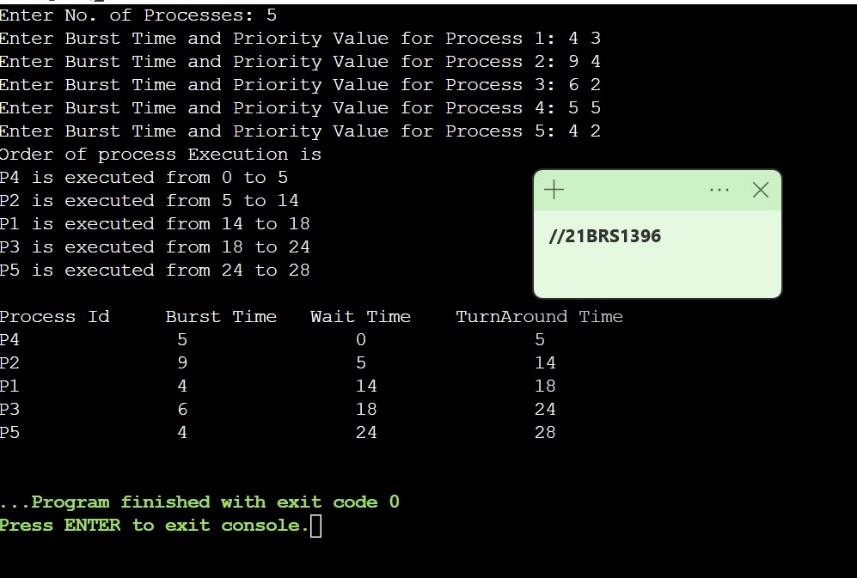
}

return 0;

}

**Output :**

21BRS1378



**Preemptive Priority: (a larger priority number implies a lower priority)**

**Code:**

#include<stdio.h> struct process

{

int WT, AT, BT, TAT, PT;

};

struct process a[10]; int main()

{

int n,temp[10],t,count=0,short\_p;

float total\_WT=0,total\_TAT=0,Avg\_WT,Avg\_TAT; printf("Enter the number of the process\n");

scanf("%d",&n);

printf("Enter the arrival time , burst time and priority of the process\n"); printf("AT BT PT\n");

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

{

scanf("%d%d%d",&a[i].AT,&a[i].BT,&a[i].PT); temp[i]=a[i].BT;

}

a[9].PT=10000;

for(t=0;count!=n;t++)

{

short\_p=9;

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

{

if(a[short\_p].PT>a[i].PT && a[i].AT<=t && a[i].BT>0)

{

short\_p=i;

}

}

a[short\_p].BT=a[short\_p].BT-1; if(a[short\_p].BT==0)

{

count++;

a[short\_p].WT=t+1-a[short\_p].AT-temp[short\_p]; a[short\_p].TAT=t+1-a[short\_p].AT; total\_WT=total\_WT+a[short\_p].WT; total\_TAT=total\_TAT+a[short\_p].TAT;

}

}

Avg\_WT=total\_WT/n;

Avg\_TAT=total\_TAT/n;

printf("Process\_Number\tWT\tTAT\n"); for(int i=0;i<n;i++)

{

printf("\tP%d\t%d\t%d\n",i+1,a[i].WT,a[i].TAT);

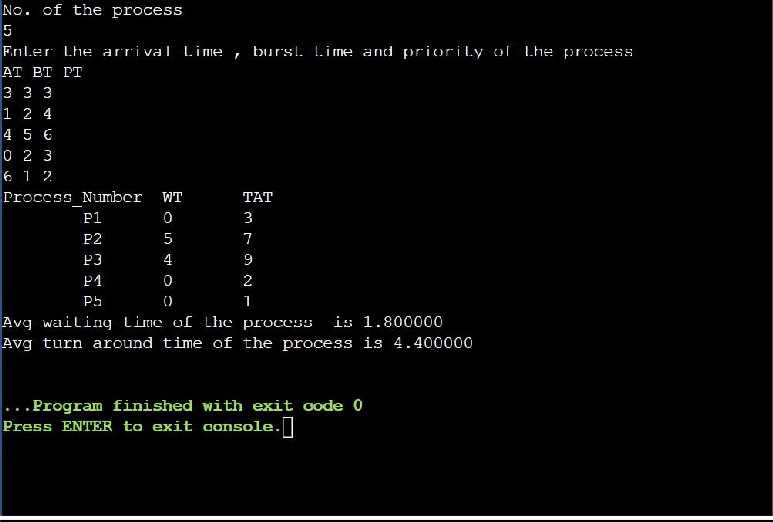
}

printf("Avg waiting time of the process is %f\n",Avg\_WT); printf("Avg turn around time of the process is %f\n",Avg\_TAT); return 0;

}

**Output:**

21BRS1378



**Round Robin (Time Quantum = 2 and all arrive at time t=0):**

**Code:**

#include<stdio.h> int main()

{

int cnt,j,n,t,remain,flag=0,tq;

int wt=0,tat=0,at[10],bt[10],rt[10]; printf("Enter Total Process:\t "); scanf("%d",&n);

remain=n; for(cnt=0;cnt<n;cnt++)

{

printf("Enter Arrival Time and Burst Time for Process Process Number %d

:",cnt+1); scanf("%d",&at[cnt]);

scanf("%d",&bt[cnt]); rt[cnt]=bt[cnt];

}

printf("Enter Time Quantum:\t"); scanf("%d",&tq);

printf("\n\nProcess\t|Turnaround Time|Waiting Time\n\n"); for(t=0,cnt=0;remain!=0;)

{

if(rt[cnt]<=tq && rt[cnt]>0)

{

t+=rt[cnt]; rt[cnt]=0; flag=1;

}

else if(rt[cnt]>0)

{

rt[cnt]-=tq; t+=tq;

}

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

{

remain--;

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

tat+=t-at[cnt]; flag=0;

}

if(cnt==n-1) cnt=0;

else if(at[cnt+1]<=t) cnt++;

else

cnt=0;

}

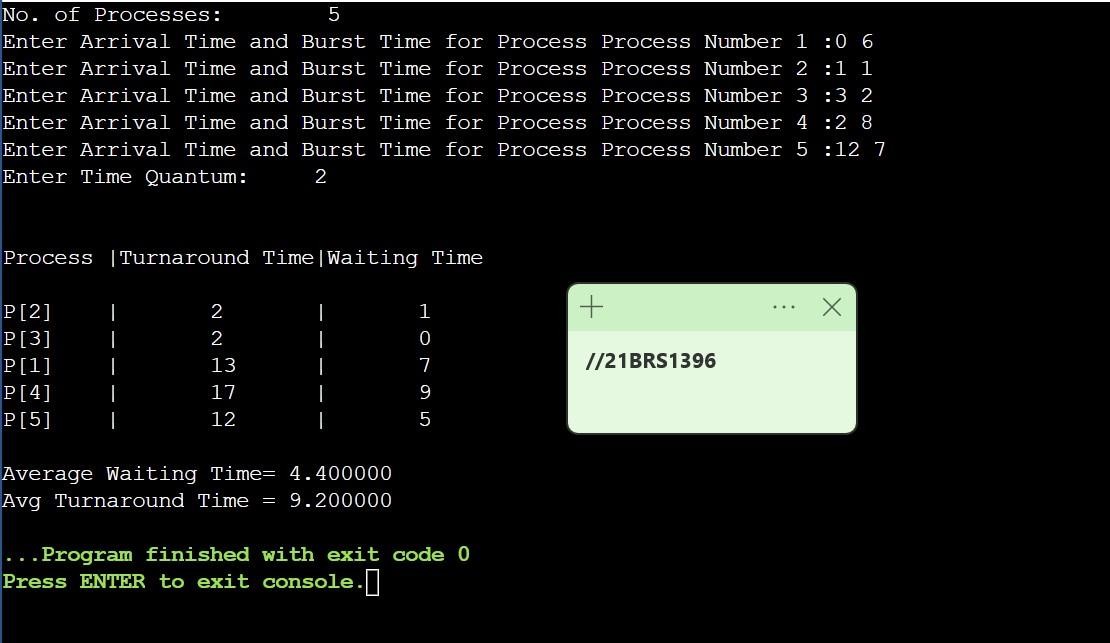
printf("\nAverage Waiting Time= %f\n",wt\*1.0/n); printf("Avg Turnaround Time = %f",tat\*1.0/n);

return 0;

}

**Output:**

21BRS1378



**Round Robin for Quantum 3:**

**Code:**

#include<stdio.h>

int main()

{

int cnt,j,n,t,remain,flag=0,tq;

int wt=0,tat=0,at[10],bt[10],rt[10]; printf("Enter Total Process:\t "); scanf("%d",&n);

remain=n; for(cnt=0;cnt<n;cnt++)

{

printf("Enter Arrival Time and Burst Time for Process Process Number %d

:",cnt+1); scanf("%d",&at[cnt]);

scanf("%d",&bt[cnt]); rt[cnt]=bt[cnt];

}

printf("Enter Time Quantum:\t"); scanf("%d",&tq);

printf("\n\nProcess\t|Turnaround Time|Waiting Time\n\n"); for(t=0,cnt=0;remain!=0;)

{

if(rt[cnt]<=tq && rt[cnt]>0)

{

t+=rt[cnt]; rt[cnt]=0; flag=1;

}

else if(rt[cnt]>0)

{

rt[cnt]-=tq; t+=tq;

}

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

{

remain--;

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

tat+=t-at[cnt]; flag=0;

}

if(cnt==n-1) cnt=0;

else if(at[cnt+1]<=t) cnt++;

else

cnt=0;

}

printf("\nAverage Waiting Time= %f\n",wt\*1.0/n); printf("Avg Turnaround Time = %f",tat\*1.0/n);

return 0;

**Output:**

21BRS1378

