**FIRST COME FIRST SERVE**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void main()

{

int bt[10],at[10],left,right,mid;

int i,n,tat,wt,stime,gt[10];

float tat\_sum,wt\_sum;

char outstr[20];

int gd=DETECT,gm;

initgraph(&gd,&gm,"C:\\turboc3\\bgi");

cleardevice();

printf("Enter no of Jobs:");

scanf("%d",&n);

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

{

printf("Arrival Time P[%d]: ",i+1);

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

printf("Burst Time P[%d]: ",i+1);

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

}

printf("\n\nProcess\tTurnAround Time \t Waiting Time\n\n");

stime=0;

wt=0;

tat=0;

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

{

gt[i]=stime;

tat=stime+bt[i]-at[i];

wt=tat-bt[i];

printf("P[%d]\t\t %d \t\t %d\n",i+1,tat,wt);

tat\_sum=tat\_sum+tat;

wt\_sum=wt\_sum+wt;

stime=stime+bt[i];

}

gt[i]=stime;

printf("\n\n Average Turn Around Time: %f ",tat\_sum/n);

printf("\n\n Average Waiting Time: %f\n ",wt\_sum/n);

getch();

cleardevice();

outtextxy(70,50,"Gantt chart");

left=0;

right=70;

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

{

sprintf(outstr,"%d",i+1);

left=left+70;

right=right+70;

mid=(left+right)/2;

rectangle(left,100,right,150);

outtextxy(mid-10,125,"P");

outtextxy(mid,125,outstr);

sprintf(outstr,"%d",gt[i]);

outtextxy(left,160,outstr);

}

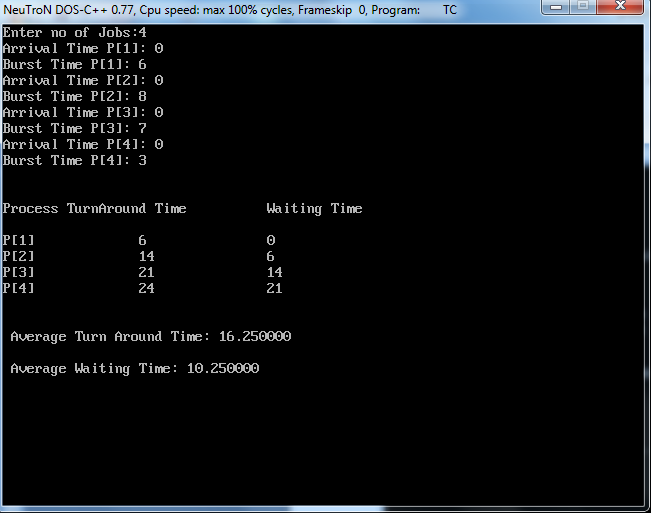
sprintf(outstr,"%d",gt[i]);

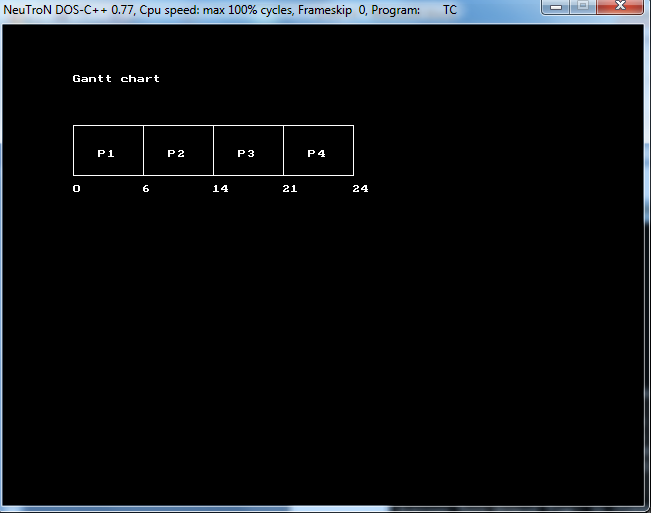
outtextxy(left+70,160,outstr);

getch();

closegraph();

}

OUTPUT:



**SHORTEST JOB FIRST**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<dos.h>

void main()

{

int st,bt[10],at[10],sum\_bt=0,sm,n,i,tat,wt;

int small[10],gt[10],count,left,right,mid;

float sum\_tat,sum\_wait;

char outstr[20];

int gd=DETECT,gm;

initgraph(&gd,&gm,"C:\\turboc3\\bgi");

cleardevice();

printf("Enter no of processes : ");

scanf("%d",&n);

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

{

printf("Arrival time for P%d : ",i+1);

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

printf("Burst time for P%d : ",i+1);

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

sum\_bt+=bt[i];

}

bt[9]=9999;

count=0;

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

for(st=0;st<sum\_bt; )

{

gt[count]=st;

sm=9;

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

{

if(at[i]<=st && bt[i]>0 && bt[i]<bt[sm])

sm=i;

}

small[count]=sm;

tat=st+bt[sm]-at[sm];

wt=tat-bt[sm];

printf("P[%d]\t\t%d\t\t%d\n",sm+1,tat,wt);

sum\_tat+=tat;

sum\_wait+=wt;

st+=bt[sm];

bt[sm]=0;

count++;

}

gt[count]=st;

printf("\n\n average WT = %f",(float)(sum\_wait/n));

printf("\n\n average TAT = %f",(float)(sum\_tat/n));

getch();

cleardevice();

outtextxy(70,50,"Gantt chart");

left=0;

right=70;

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

{

sprintf(outstr,"%d",small[i]+1);

left=left+70;

right=right+70;

mid=(left+right)/2;

rectangle(left,100,right,150);

outtextxy(mid-10,125,"P");

outtextxy(mid,125,outstr);

sprintf(outstr,"%d",gt[i]);

outtextxy(left,160,outstr);

}

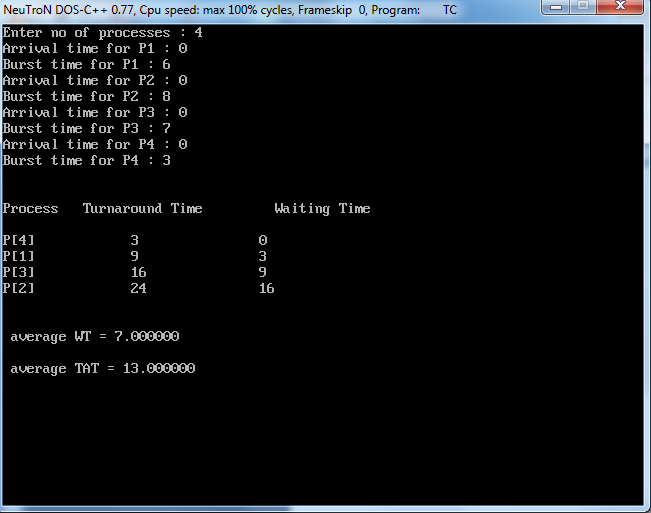
sprintf(outstr,"%d",gt[i]);

outtextxy(left+70,160,outstr);

getch();

closegraph();

}



OUTPUT:

