**Program**

#include<stdio.h>

#include<stdlib.h>

void main()

{

int i,n,ch,j,t1,t2,t3;

float avw,avt;

do

{

printf("Enter choice: 1.FCFS 2.SJF 3.Priority Scheduling 4.Round robin Scheduling 5.Exit\n");

scanf("%d",&ch);

switch(ch)

{

case 1:

{

printf("Enter number of processes:");

scanf("%d",&n);

int p[n],bt[n],wt[n],tat[n],tw=0,tt=0;

printf("Enter processes numbers:");

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

{

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

}

printf("Enter burst time for each processes:");

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

{

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

}

wt[0]=0;

tat[0]=bt[0];

tw=tw+wt[0];

tt=tt+tat[0];

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

{

wt[i]=wt[i-1]+bt[i-1];

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

tw=tw+wt[i];

tt=tt+tat[i];

}

avw=tw/(float)n;

avt=tt/(float)n;

printf("Gantt Chart:\n");

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

printf("\t\tP%d",p[i]);

printf("\n");

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

printf("\t%d\t",wt[i]);

printf("\t%d\n",tat[n-1]);

printf("Process\tburst time\twaiting time\tturn around time\n");

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

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

printf("Total waiting time=%dms\n",tw);

printf("Total turn around time=%dms\n",tt);

printf("Average waiting time=%fms\n",avw);

printf("Average turn around time=%fms\n",avt);

break;

}

case 2:

{

printf("Enter number of processes:");

scanf("%d",&n);

int p[n],bt[n],wt[n],tat[n],tw=0,tt=0;

printf("Enter processes numbers:");

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

{

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

}

printf("Enter burst time for each processes:");

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

{

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

}

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

{

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

{

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

{

t1=p[i];

p[i]=p[j];

p[j]=t1;

t2=bt[i];

bt[i]=bt[j];

bt[j]=t2;

}

}

}

wt[0]=0;

tat[0]=bt[0];

tw=tw+wt[0];

tt=tt+tat[0];

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

{

wt[i]=wt[i-1]+bt[i-1];

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

tw=tw+wt[i];

tt=tt+tat[i];

}

avw=tw/(float)n;

avt=tt/(float)n;

printf("Gantt Chart:\n");

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

printf("\t\tP%d",p[i]);

printf("\n");

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

printf("\t%d\t",wt[i]);

printf("\t%d\n",tat[n-1]);

printf("Process\tburst time\twaiting time\tturn around time\n");

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

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

printf("Total waiting time=%dms\n",tw);

printf("Total turn around time=%dms\n",tt);

printf("Average waiting time=%fms\n",avw);

printf("Average turn around time=%fms\n",avt);

break;

}

case 3:

{

printf("Enter number of processes:");

scanf("%d",&n);

int p[n],bt[n],wt[n],tat[n],tw=0,tt=0,pr[n];

printf("Enter processes numbers:");

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

{

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

}

printf("Enter burst time for each processes:");

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

{

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

}

printf("Enter priority for each processes:");

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

{

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

}

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

{

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

{

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

{

t1=p[i];

p[i]=p[j];

p[j]=t1;

t2=bt[i];

bt[i]=bt[j];

bt[j]=t2;

t3=pr[i];

pr[i]=pr[j];

pr[j]=t3;

}

}

}

wt[0]=0;

tat[0]=bt[0];

tw=tw+wt[0];

tt=tt+tat[0];

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

{

wt[i]=wt[i-1]+bt[i-1];

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

tw=tw+wt[i];

tt=tt+tat[i];

}

avw=tw/(float)n;

avt=tt/(float)n;

printf("Gantt Chart:\n");

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

printf("\t\tP%d",p[i]);

printf("\n");

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

printf("\t%d\t",wt[i]);

printf("\t%d\n",tat[n-1]);

printf("Process\tburst time\tpriority\twaiting time\tturn around time\n");

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

printf("P%d\t%d\t\t%d\t\t%d\t\t%d\n",p[i],bt[i],pr[i],wt[i],tat[i]);

printf("Total waiting time=%dms\n",tw);

printf("Total turn around time=%dms\n",tt);

printf("Average waiting time=%fms\n",avw);

printf("Average turn around time=%fms\n",avt);

break;

}

case 4:

{

printf("Enter number of processes:");

scanf("%d",&n);

int p[n],bt[n],wt[n],tat[n],tw=0,tt=0,tq,tqi=0,rem[n],count=0;

printf("Enter processes numbers:");

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

{

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

}

printf("Enter time quantum:");

scanf("%d",&tq);

printf("Enter burst time for each processes:");

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

{

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

rem[i]=bt[i];

}

printf("Gantt Chart:\n");

int a[50],k=1;

a[0]=0;

while(count<n)

{

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

{

if(rem[i]>0)

{

if(rem[i]>tq)

{

printf("\t\tP%d",p[i]);

tqi=tqi+tq;

rem[i]=rem[i]-tq;

}

else

{

printf("\t\tP%d",p[i]);

tqi=tqi+rem[i];

rem[i]=0;

wt[i]=tqi-bt[i];

tat[i]=tqi;

count++;

}

a[k]=tqi;

k++;

}

}

}

printf("\n");

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

printf("\t%d\t",a[i]);

printf("\n");

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

{

tw=tw+wt[i];

tt=tt+tat[i];

}

avw=tw/(float)n;

avt=tt/(float)n;

printf("Process\tburst time\twaiting time\tturn around time\n");

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

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

printf("Total waiting time=%dms\n",tw);

printf("Total turn around time=%dms\n",tt);

printf("Average waiting time=%fms\n",avw);

printf("Average turn around time=%fms\n",avt);

break;

}

case 5:

{

exit(0);

}

default:

{

printf("error");

break;

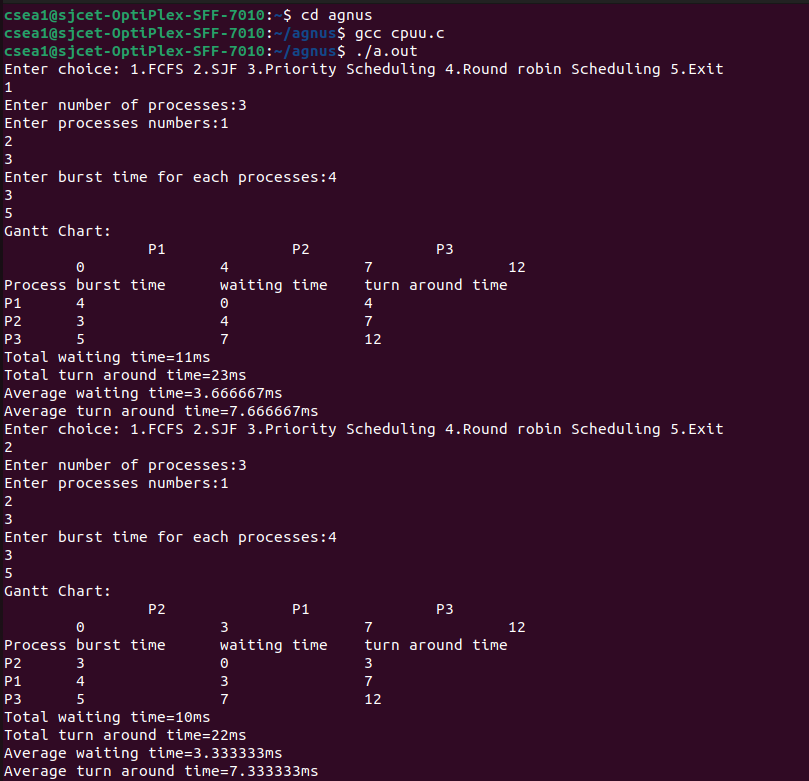
}

}

}while(ch!=0);

}

**Output**

****

