**PROGRAM:**

#include<stdio.h>

int main()

{

int pr[10],p[10],bt[10],wt[10],tat[10],i,j,n,pos,temp;

float at=0,aw=0;

printf("PRIORITY");

printf("\n enter number of processess:");

scanf("%d",&n);

printf("\n enter burst time and priority for each process");

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

{

printf("\nP%d\n",i);

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

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

p[i]=i;

}

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

{

pos=i;

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

{

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

pos=j;

}

temp=pr[i];

pr[i]=pr]pos];

pr[pos]=temp;

temp=bt[i];

bt[i]=bt[pos];

bt[pos]=temp;

temp=p[i];

p[i]=p[pos];

p[pos]=temp;

}

wt[1]=0;

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

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

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

aw+=wt[i];

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

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

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

at+=tat[i];

printf("\n process burst time waiting time turnaround time");

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

{

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

}

printf("\n average waiting time=%f",aw/n);

printf("\n average turnaround time=%f",at/n);

printf("\n Grantt chart is \n");

printf("\n----------------------\n");

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

{

printf("P%d",p[i]);

}

printf("\n----------------------\n");

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

{

printf("%d",tat[i]);

}

return 0;

}

**OUTPUT:**

PRIORITY

Enter number of processess : 4

Enter burst time for each process :

P1 6 3

P2 2 2

P3 14 1

P4 6 4

Process Burst time Waiting time Turnaround time

P3 14 0 14

P2 2 14 16

P1 6 16 22

P4 6 22 28

Average Waiting Time = 13.000000

Average Turnaround time = 20.000000

Grantt chart is

------------------

|P3 | P2 | P1 | P4 |

0 14 16 22 28

PROGRAM:

#include<stdio.h>

int main()

{

int count,j,n,time,remain,flag=0,tq,i,b,p=0;

int wait\_time=0,turnaround\_time=0,at[10],bt[10],rt[10],c[20]={0};

int br[10],wo[10];

printf(“ROUND ROBIN\n\n”);

printf(“enter total process”);

scanf(“%d”,&n);

remain=n;

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

{

printf(“enter burst time process time process number %d”,count+1);

at[count]=0;

scanf(“%d”,&bt[count]);

rt[count]=bt[count];

}

printf(“enter time quantum”);

scanf(“%d”,&tq);

printf(“\n\n process burst time waiting time turnaround time \n\n”);

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

{

If(rt[count]<=tq && rt[count]>0)

{

time+=rt[count];

rt[count]=0;

flag=1;

}

elseif (rt[count]>0)

{

rt[count]-=tq;

time+=tq;

}

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

{

remain--;

printf(“P[%d] %d %d %d \n”,count+1),bt[count],time-at[count]-bt[count],time-at[count]);

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

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

flag=0;

}

if(count==n-1)

count=0;

elseif(at[count+1]<=time)

count++;

else

count=0;

}

printf(“\n average waiting time=%f\n”,wait\_time\*1.0/n);

printf(“\n average turnaround time = %f”,turn around\_time\*1.0/n);

for(i=0;i<10;i++)

wo[i]=0;

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

br[i]=bt[i];

b=0;

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

b=b+bt[i];

count=0;

printf(“grant chart is:\ --------------------------------------\n”);

do

{

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

{

if(br[i]==0)

{

}

else

{

if(br[i]>=tq)

{

br[i]=br[i]-tq;

if(br[i]==0)

wo[i]=count;

count=count+tq;

c[p]=count;

p++;

printf(“P%d”,i+1);

}

else

{

if(br[i]<tq)

{

count=count+br[i];

br[i]=0;

wo[i]=count;

c[p]=count;

p++;

printf(“P%d”,i+1);

}

}

}

}

}

while(count<b);

printf(“\n \n ---------------------------“);

printf(“\n\n”);

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

printf(“%d”,c[i]);

return 0;

}

OUTPUT:

ROUND ROBIN

Enter total process : 4

Enter burst time for process number 1 : 9

Enter burst time for process number 2 : 5

Enter burst time for process number 3 : 3

Enter burst time for process number 4 : 4

Enter time quantum : 5

Process Burst time Waiting time Turnaround time

P[2] 5 5 10

P[3] 3 10 13

P[4] 4 13 17

P[1] 9 12 21

Average waiting time = 10.000000

Average turnaround time = 15.250000