rr\_07.cpp

**Compile:** g++ rr\_07.cpp -o rr\_07

**Run:** ./rr\_07

**Program:**

#include<iostream>

using namespace std;

classsched

{

public:

intn,bt[10],at[10],tat[10],wt[10],rt[10],finish[10],twt,ttat,total;

voidreadData();

voidInit();

voiddispTime();

intgetNextProcess(int);

voidcomputeRR();

};

voidsched::readData()

{

cout<<"Enter no. of processes\n";

cin>>n;

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

{

cout<<"Enter the burst times for P["<<(i+1)<<"]: ";

cin>>bt[i];

}

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

{

cout<<"Enter the arrival times for P["<<(i+1)<<"]: ";

cin>>at[i];

}

}

voidsched::Init(){

total=0;

twt=0;

ttat=0;

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

{

rt[i]=bt[i];

finish[i]=0;

wt[i]=0;

tat[i]=0;

total+=bt[i];

}

}

voidsched::dispTime()

{

cout<<"Processes Waiting Time Turnaround Time"<<endl;

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

{

twt+=wt[i];

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

ttat+=tat[i];

cout<<"P["<<(i+1)<<"] "<<wt[i]<<" "<<tat[i]<<endl;

}

cout<<endl;

cout<<"Avg Waiting time = "<<(double)twt<<" and Avg Turnaround time = "<<(double)ttat<<endl;

cout<<endl;

cout<<"Scheduling complete....!!\n";

}

intsched::getNextProcess(int time){

inti,low;

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

if(finish[i]==0){low=i; break; }

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

if(finish[i]!=1)

if(rt[i]<rt[low] && at[i]<=time)

low=i;

return low;

}

voidsched::computeRR(){

readData();

Init();

inttime,j,q,i,dec=0;

cout<<"Enter the time quantum:\n";

cin>>q;

cout<<"Gantt Chart:\n ";

for(time=0;time<total;)

{

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

{

if(at[i]<=time && finish[i]==0)

{

cout<<"("<<time<<")|==P"<<(i+1)<<"==|";

if(rt[i]<q) {

dec=rt[i];

}

else {dec=q;}

rt[i]=rt[i]-dec;

if(rt[i]==0) finish[i]=1;

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

if(j!=i&& finish[j]==0 && at[j]<=time)

wt[j]+=dec;

time=time+dec;

}

}

}

cout<<"("<<total<<")"<<endl;

dispTime();

cout<<endl;

}

int main()

{

sched s;

s.computeRR();

}

**Output:**

