#include<stdio.h>

int count,i,j,time,remain,flag=0,time\_quantum;

int wait\_time=0,turnaround\_time=0,fa[10],fb[10],rt[10],sa[10],sb[10];

int f,s,p;

int ra[15],rb[15];

int fname[15],sname[15],rname[15];

int input(int \*a, int \*b,int \*c)

{

int n;

printf("Enter Total Process:\t ");

scanf("%d",&n);

remain=n;

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

{

printf("Enter the name: ");

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

printf("Enter Arrival Time and Burst Time for Process Process Number %d :",count+1);

scanf("%d",&a[count]);

scanf("%d",&b[count]);

printf("\n");

}

return n;

}

int round\_robin()

{

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

{

rt[i]=rb[i];

}

remain=p;

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

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

{

if(rt[count]<=time\_quantum && rt[count]>0)

{

time+=rt[count];

rt[count]=0;

flag=1;

}

else if(rt[count]>0)

{

rt[count]-=time\_quantum;

time+=time\_quantum;

}

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

{

remain--;

printf("P[%d]\t|\t%d\t|\t%d\n",rname[count],time-ra[count],time-ra[count]-rb[count]);

wait\_time+=time-ra[count]-rb[count];

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

flag=0;

}

if(count==p-1)

count=0;

else if(ra[count+1]<=time)

count++;

else

count=0;

}

printf("\nAverage Waiting Time= %f\n",wait\_time\*1.0/p);

printf("Avg Turnaround Time = %f",turnaround\_time\*1.0/p);

return 0;

}

int sort()

{

int swap;

char sw[15];

for (i=0;i<p-1;i++)

{

for (j=0;j< p-i-1;j++)

{

if (ra[j] > ra[j+1])

{

swap = ra[j];

ra[j]= ra[j+1];

ra[j+1]= swap;

swap = rb[j];

rb[j]= rb[j+1];

rb[j+1]= swap;

swap = rname[j];

rname[j]= rname[j+1];

rname[j+1]= swap;

}

}

}

}

int sortedMerge()

{

int i = 0, j = 0, k = 0;

while (i < f) {

ra[k] = fa[i];

rb[k] = fb[i];

rname[k]=fname[i];

i += 1;

k += 1;

}

while (j < s) {

ra[k] = sa[j];

rb[k] = sb[j];

rname[k] = sname[j];

j += 1;

k += 1;

}

p=k;

sort();

}

int main()

{

printf("Enter data for FACULTY\n");

f=input(fa,fb,fname);

printf("Enter data for STUDENT\n");

s=input(sa,sb,sname);

printf("Enter Time Quantum:\t");

scanf("%d",&time\_quantum);

sortedMerge();

round\_robin();

}