**NAME :** P.Rohith Sharma **Registration No**: 11802975

**Roll Number :**11 **Section :** K18BY (**Group-1**)

**Email Address :** sharmarohithprk10@gmail.com

**Code :** 3rd Question

**Problem :**

If a teacher is being served at the food mess and during the period when he is being served, another teacher comes, then that teacher would get the service (food) next. This process might continue leading to increase in waiting time of students to get food. Propose an algorithm and implement it using a C program to ensure average waiting time of students is minimized and that students do not get starved.

**Solution :**

#include<stdio.h>

int main()

{

int count,j,n,time,rest\_time;

int flag=0,time\_quantum;

int wait\_time=0,turnaround\_time=0,at[10],bt[10],rt[10];

printf("EnterTotalProcess:\t");

scanf("%d",&n);

rest\_time=n;

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

{

printf("EnterArrivalTimeandBurstTimeforProcessProcess Number%d:\n",count+1);

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

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

rt[count]=bt[count];

}

printf("EnterTimeQuantum:\t");

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

printf("\n\nProcess\t|TurnaroundTime|WaitingTime\n\n");

for(time=0,count=0;rest\_time!=0;)

{

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

{

time+=rt[count];

rt[count]=0;

flag=1;

}

elseif(rt[count]>0)

{

rt[count]-=time\_quantum;

time+=time\_quantum;

}

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

{

rest\_time--;

printf("P[%d]\t|\t%d\t|\t%d\n",count+1,time-at[count],time-at[coun t]-bt[count]);

wait\_time+=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("\nAverageWaitingTime=%f\n",wait\_time\*1.0/n);

printf("AvgTurnaroundTime=%f",turnaround\_time\*1.0/n);

return0;

}

**OUTPUT :**

**TEST CASE-1 :**



**TEST CASE-2 :**

