Question no 21

#include<stdio.h>

struct ab{

int a;

};

int fifo(int page [3],int frames\_no[10])

{

int i;

for(i=3;i<10;i++)

printf("\n %d \t%d ",page[i], frames\_no[i]);

}

int lru(int page[3],int frames\_no[10])

{

printf("\nfifgho");

}

int optimal(int page[3],int frames\_no[10])

{

printf("\nfikhjfo");

}

int main()

{

float avg=0,avg1=0;

int i,j;

int frames\_no [10],page[3];

printf("\n enter 10 demands");

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

scanf("%d",&frames\_no[i]);

for(i=0;i<3;i++){

if (i==0){

// page is blank

page[i]=frames\_no[i];

avg=((0.7\*8)+(.3\*100));

printf("\npage is blanck page acesss time%f", avg);

}

else {

// not replaced

avg=((0.7\*8)+(.3\*100));

printf("\npage fault not replaced %f", avg);

}

}

for(i=3;i<10;i++){

for(j=0;j<3;j++)

{

if(frames\_no[i]==page[j]){

printf("\n no page fault");

break;

}

else

j++;

}

if(page[j]!=frames\_no[i]){

avg1=((0.7\*20)+(.3\*100));

printf("\npage fault page replaced %f ",avg1);

}

}

}

Question 10

#include <stdio.h>

#include<conio.h>

struct t{

int a,b;

};

struct t1{

int c, d;

};

struct t shortest\_time(int limit,int arrival\_time[10],int burst\_time[10],int temp[10])

{

struct t re;

int i, smallest, count = 0, time;

double wait\_time = 0, turnaround\_time = 0, end;

float average\_waiting\_time, average\_turnaround\_time;

burst\_time[9] = 9999;

for(time = 0; count != limit; time++)

{

smallest = 9;

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

{

if(arrival\_time[i] <= time && burst\_time[i] < burst\_time[smallest] && burst\_time[i] > 0)

{

smallest = i;

}

}

burst\_time[smallest]--;

if(burst\_time[smallest] == 0)

{

count++;

end = time + 1;

wait\_time = wait\_time + end - arrival\_time[smallest] - temp[smallest];

turnaround\_time = turnaround\_time + end - arrival\_time[smallest];

}

}

re.a= average\_waiting\_time = wait\_time / limit;

re.b= average\_turnaround\_time = turnaround\_time / limit;

// printf("\n\nAverage Waiting Time:\t%lf\n", average\_waiting\_time);

// printf("Average Turnaround Time:\t%lf\n", average\_turnaround\_time);

return re;

}

struct t1 round\_robin(int limit,int at[10],int bt[10],int rt[10])

{

struct t1 ret;

int count,j,time,remain=limit,flag=0;

int wait\_time=0,turnaround\_time=0;

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

{

if(rt[count]<=2 && rt[count]>0)

{

time+=rt[count];

rt[count]=0;

flag=1;

}

else if(rt[count]>0)

{

rt[count]-=2;

time+=2;

}

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

{

remain--;

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

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

flag=0;

}

if(count==limit-1)

count=0;

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

count++;

else

count=0;

}

ret.c=wait\_time\*1.0/limit;

ret.d=turnaround\_time\*1.0/limit;

return ret;

}

int main()

{

int i,limit;

float oo,ooo;

printf("\nEnter the Total Number of Processes:\t");

scanf("%d", &limit);

int arrival\_time[limit], burst\_time[limit],temp[limit],rt[10];

printf("\nEnter Details of %d Processes\n", limit);

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

{

printf("\nEnter Arrival Time:\t");

scanf("%d", &arrival\_time[i]);

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

scanf("%d", &burst\_time[i]);

temp[i] = burst\_time[i];

rt[i]=burst\_time[i];

}

struct t n=shortest\_time(limit,arrival\_time, burst\_time,temp);

struct t1 n1= round\_robin(limit, arrival\_time, burst\_time,rt);

oo=n.a+n1.c;

ooo=n.b+n1.d;

printf("Avg Turnaround Time = %f", oo);

printf("\nAverage Waiting Time= %f\n",ooo);

}