Operating Systems Lab

Endsemester Exam

Codes Part

-Maanas Bhardwaj

-AU1940047

-Section-2

Question-2-a

```
#include<stdio.h>
int main(){

        int i, PN, total = 0, x, counter = 0, time_quantum;
        int wait_time = 0, turnaround_time = 0, arrival_time[10], burst_time[10], temp[10];
        float average_wait_time, average_turnaround_time;

        printf("\nEnter Total Number of Processes:  ");
        scanf("%d", &PN);
        x =PN;

        for(i = 0; i < PN; i++){
                printf("\nEnter Details of Process[%d]\n", i + 1);
                printf("Arrival Time:\t");
                scanf("%d", &arrival_time[i]);
                printf("Burst Time:\t");
                scanf("%d", &burst_time[i]);
                temp[i] = burst_time[i];
        }

        printf("\nEnter Time Quantum:");
        scanf("%d", &time_quantum);
        printf("\nProcess ID  Burst Time Turnaround Time Waiting Time\n");

        for(total = 0, i = 0; x != 0;)
        {
                if(temp[i] <= time_quantum && temp[i] > 0)
                {
                        total = total + temp[i];
                        temp[i] = 0;
                        counter = 1;
                I
                }
                else if(temp[i] > 0)
                {
                        temp[i] = temp[i] - time_quantum;
                        total = total + time_quantum;
                }
                if(temp[i] == 0 && counter == 1)
                {
                        x--;
                        printf("\nProcess[%d]  \t%    %d      %d", i + 1, burst_time[i], total - arrival_time[i], total-arrival_time[i] - burst_time[i] );
                        wait time = wait time + total - arrival_time[i] - burst_time[i];
                        turnaround_time = turnaround_time + total - arrival_time[i];
                        counter = 0;
                }
                if(i == PN - 1)
                {
                        i = 0;
                }
                else if(arrival_time[i + 1] <= total)
        {
            i++;
        }
        else
        {
            i=0;
        }
    }

    average_wait_time = wait_time* 1.0/PN;
    average_turnaround_time= turnaround_time*1.0/PN;
    printf("\nAverage Waiting Time: %f", average_wait_time);
    printf("\nAvg Turnaround Time: %f\d", average_turnaround_time);
    return 0;
}
```

Note-

I was not be able to write the code for Modified RR but the explaination of the pseudo code is done in the description part

Question3

Pseudocode for Producer-Consumer Problem

```
Variable = 100
Sem full, empty
Full = 0
Empty= n


Wait(Sem s){
s--
}
Signal(Sem s){
s++
}


Process-1
Proceducer{
    While(<condition>){
        //Assigning values to buffer by doing some task
        }
        Wait(empty);            //Filling the buffer
        Wait(Variable);

        Signal(variable);
        Signal(full);

}

Consumer{
    While(<condition>){
        //Assigning values to buffer by doing some task
        }
        Wait(full);                 //Emptying the buffer
        Wait(Variable);


        Signal(variable);
        Signal(emptly);


}
```