# Name: Mohd Adil

# Roll No: 20BCS042

**A program to implement the First Come First Serve scheduling algorithm and find the average turnaround time, waiting time, completion time and**

**response time for overall process. Also Printing Gantt chart for it.**

//FCFS

#include<iostream>

using namespace std;

int n;

float avgCt, avgWt, avgTt;

struct Process{

    char Pname[5];

    int arvlTime;

    int brstTime

int cmpTime;

    int wtngTime

int tatTime;

    struct Process \*next;

};

int isEmpty(Process \*front){

    if(front==NULL || n==0){

        return 1;

    }

    return 0;

}

struct Process \*insert(Process \*front, int i){

    struct Process \*p = (struct Process\*)malloc(sizeof(struct Process));

    cout<<"Enter the name of the Process "<<i<<", its Burst and Arrival Time : ";

    cin>>p->Pname>>p->brstTime>>p->arvlTime;

    p->next = NULL;

    if(front==NULL){

        front = p;

    }

    else if (front->arvlTime > p->arvlTime){

        p->next = front;

        front = p;

    }

    else{

        struct Process \*tmp = front;

        while (tmp->next != NULL && tmp->next->arvlTime < p->arvlTime){

            tmp = tmp->next;

        }

        p->next = tmp->next;

        tmp->next = p;

    }

    return front;

}

void calculate(Process \*front){

    if(isEmpty(front)){

        cout<<"\nNo processes in the ready Queue";

        return;

    }

    front->wtngTime=0;

    front->cmpTime=front->brstTime;

    //calculating completion time

    int prv = front->cmpTime;

    struct Process \*tmp = front->next;

    while(tmp!=NULL){

        tmp->cmpTime = prv + tmp->brstTime;

        prv = tmp->cmpTime;

        tmp=tmp->next;

    }

    //calculating waiting time

    prv = front->cmpTime;

    tmp = front->next;

    while(tmp!=NULL){

        tmp->wtngTime = prv - tmp->arvlTime;

        prv = tmp->cmpTime;

        tmp=tmp->next;

    }

    //calculating turn arround time

    tmp = front;

    while(tmp!=NULL){

        tmp->tatTime = tmp->wtngTime + tmp->brstTime;

        tmp=tmp->next;

    }

    //calculating average time

    tmp = front;

    float s1=0, s2=0, s3=0;

    while(tmp!=NULL){

        s1 = s1 + tmp->cmpTime;

        s2 = s2 + tmp->wtngTime;

        s3 = s3 + tmp->tatTime;

        tmp=tmp->next;

    }

    avgCt = s1/n;

    avgWt = s2/n;

    avgTt = s3/n;

}

void display(Process \*front){

    if(isEmpty(front)){

        cout<<"\nNo processes in the ready Queue";

        return;

    }

    cout<<"\n\nDisplaying the table :- ";

    struct Process \*tmp = front;

    cout<<"\n\n+--------------+------------+--------------+-----------------+--------------+-----------------+---------------+";

    cout<<"\n| Process name | Burst Time | Arrival Time | Completion Time | Waiting Time | TurnAround Time | Response Time |";

    cout<<"\n+--------------+------------+--------------+-----------------+--------------+-----------------+---------------+";

    while(tmp!=NULL){

        printf("\n|      %s      |    %2d      |      %2d      |        %2d       |      %2d      |      %2d         |      %2d       |"

               ,tmp->Pname, tmp->brstTime, tmp->arvlTime, tmp->cmpTime, tmp->wtngTime, tmp->tatTime, tmp->wtngTime);

    cout<<"\n+--------------+------------+--------------+-----------------+--------------+-----------------+---------------+";

        tmp=tmp->next;

    }

    cout<<"\n\n";

    printf("\nAverage Completion time : %.2fns", avgCt);

    printf("\nAverage Waiting time : %.2fns", avgWt);

    printf("\nAverage TurnAround time : %.2fns", avgTt);

    printf("\nAverage Response time : %.2fns", avgWt);

}

void printGanttChart(Process \*front){

    if(isEmpty(front)){

        cout<<"\nNo processes in the ready Queue";

        return;

    }

    cout<<"\n\nGantt Chart : ";

    struct Process \*tmp = front;

    cout<<"\n\n+";

    while(tmp!=NULL){

        for(int i=0; i<2\*tmp->brstTime; i++){

            cout<<"-";

        }

        cout<<"+";

        tmp = tmp->next;

    }

    tmp = front;

    cout<<"\n|";

    while(tmp!=NULL){

        for(int i=0; i<tmp->brstTime-1; i++){

            cout<<" ";

        }

        cout<<tmp->Pname;

        for(int i=0; i<tmp->brstTime-1; i++){

            cout<<" ";

        }

        cout<<"|";

        tmp = tmp->next;

    }

    tmp = front;

    cout<<"\n+";

    while(tmp!=NULL){

        for(int i=0; i<2\*tmp->brstTime; i++){

            cout<<"-";

        }

        cout<<"+";

        tmp = tmp->next;

    }

    tmp = front;

    cout<<"\n0";

    while(tmp!=NULL){

        for(int i=0; i<2\*tmp->brstTime-1; i++){

            cout<<" ";

        }

        // cout<<tmp->cmpTime;

        printf("%2d", tmp->cmpTime);

        tmp = tmp->next;

    }

    cout<<"\n\n";

}

int main(){

    cout<<"\nName : Mohd Adil";

    cout<<"\nRoll No : 20BCS042";

    cout<<"\nEnter the number of process";

    cin>>n;

    struct Process \*front = NULL;

    for(int i=1; i<=n; i++){

        front = insert(front,i);

    }

    calculate(front);

    display(front);

    printGanttChart(front);

return 0;

}

// 5 P1 6 2 P2 2 5 P3 8 1 P4 3 0 P5 4 4

# Output:

