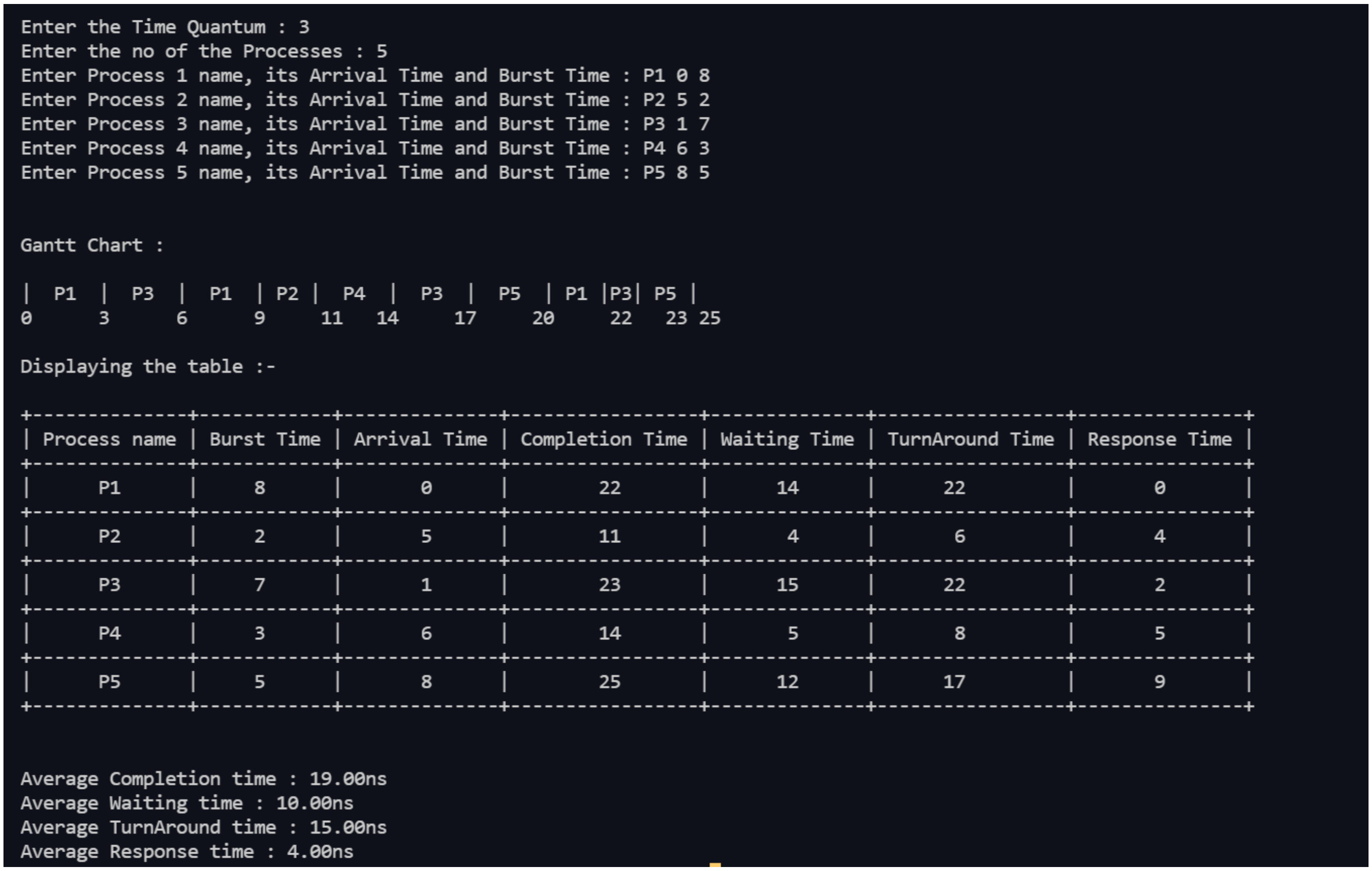
Name : Mohd Adil

Roll No : 20BCS042

A program to implement the Round Robin scheduling algorithm with time quantum =t and find the average turnaround time, waiting time, completion time and response time for the overall process. Also Printing Gantt chart for it.

```
#include<iostream>
#include<vector>
#include<queue>

using namespace std;

struct Process{
    char Pname[3];
    int id;
    int Times[6];
    vector<pair<int,int>> SEtime;
    //for calculations
    int b;
};

vector<Process> v;
vector<pair<int, int>> TimeSet;
int n;
float avgc, avgw, avgt, avgr;

void printProcess(string pname, int s, int e){
    TimeSet. push_back(make_pair (s, e));
    cout<<"|";
    int block = e-s;
    for(int i=0; i<e-s-1; i++) cout<<" ";
    cout<<pname;
    for(int i=0; i<e-s-1; i++) cout<<" ";
}

void printGantt(){
    cout<<"0";
    cout<<"     ";
    for(auto T:TimeSet){
        printf("%2d", T.second);
        for(int i=0; i<2*(T.second-T.first)-1; i++) cout<<" ";
    }
}

void RoundRobin(int TimeQuantum){
    queue<Process> pq;
    pq.push(v.front());

    int currentTime=0;
    bool visited[n] = {false};
    visited[0]=true;

    while(!pq.empty()){
        Process p = pq.front();
        pq.pop();
        int Pid = p.id;
        int tb = min(TimeQuantum, p.b);

        pair<int, int> pr;
        pr.first = currentTime;
        currentTime+=tb;
        pr.second = currentTime;
        v[Pid].SEtime.push_back(pr);
        v[Pid].b-=tb;
        // cout<<v[Pid].Pname<<" "<<v[Pid].b<<endl;
        printProcess(p.Pname, pr.first, pr.second);

        for(int i=0; i<v.size(); i++){
            if(v[i].Times[0]<=currentTime && v[i].b!=0 && !visited[v[i].id]){
                // cout<<v[i].Pname<<" ";
                pq.push(v[i]);
                visited[v[i].id]=true;
            }
        }
        if(v[Pid].b!=0){
            pq.push(v[Pid]);
        }
    }
    cout<<"|";
    cout<<endl;
    printGantt();
}

void calculateTimes(){

    float sumc=0, sumw=0, sumt=0, sumr=0;

    //calculating completion time
    for(auto &p : v){
        int sze = p.SEtime.size();
        p.Times[2] = p.SEtime[sze-1].second;
        sumc += p.Times[2];
    }

    //calculating turn around time
    // CT-AR
    for(auto &p : v){
        p.Times[4] = p.Times[2] - p.Times[0];
        sumt += p.Times[4];
    }


    //calculating waiting time
    // TAT-BT
    for(auto &p : v){
        p.Times[3] = p.Times[2] - p.Times[0] - p.Times[1];
        sumw += p.Times[3];
    }


    //calculating Response Time
    // First - AT
    for(auto &p : v){
        p.Times[5] = p.SEtime[0].first - p.Times[0];
        sumr += p.Times[5];
    }

    //calculating avg(s) time
    avgc = sumc/n;
    avgw = sumw/n;
    avgt = sumt/n;
    avgr = sumr/n;
}

void display(){
    cout<<"\n\nDisplaying the table :- ";

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

    for(auto i:v){
        printf("\n|     %s     |    %2d     |     %2d     |      %2d      |     %2d     |     %2d       |     %2d      |"
               ,i.Pname, i.Times[1], i.Times[0], i.Times[2], i.Times[3], i.Times[4], i.Times[5]);
    cout<<"\n+--------------+------------+--------------+------------------+--------------+------------------+---------------+";
    }

    cout<<"\n\n";
    printf("\nAverage Completion time : %.2fns", avgc);
    printf("\nAverage Waiting time : %.2fns", avgw);
    printf("\nAverage TurnAround time : %.2fns", avgt);
    printf("\nAverage Response time : %.2fns", avgr);
}

int main(){
    int TimeQuantum;
    cout<<"Enter the Time Quantum : ";
    cin>>TimeQuantum;
    cout<<"Enter the no of the Processes : ";
    cin>>n;
    for(int i=0; i<n; i++){
        struct Process p;
        cout<<"Enter Process "<<i+1<<" name, its Arrival Time and Burst Time : ";
        cin>>p.Pname>>p.Times[0]>>p.Times[1];
        p.id=i;
        p.b = p.Times[1];
        v.push_back(p);
    }
    cout<<endl<<endl<<"Gantt Chart : "<<endl<<endl;
    RoundRobin(TimeQuantum);
    calculateTimes();
    display();
    return 0;
}
```

Output :



```
Enter the Time Quantum : 3
Enter the no of the Processes : 5
Enter Process 1 name, its Arrival Time and Burst Time : P1 0 8
Enter Process 2 name, its Arrival Time and Burst Time : P2 5 2
Enter Process 3 name, its Arrival Time and Burst Time : P3 1 7
Enter Process 4 name, its Arrival Time and Burst Time : P4 6 3
Enter Process 5 name, its Arrival Time and Burst Time : P5 8 5


Gantt Chart :

|  P1  |  P3  |  P1  | P2 |  P4  |  P3  |  P5  | P1 |P3| P5 |
0      3      6      9     11   14     17     20     22   23 25

Displaying the table :-

+--------------+------------+--------------+-----------------+--------------+-----------------+---------------+
| Process name | Burst Time | Arrival Time | Completion Time | Waiting Time | TurnAround Time | Response Time |
+--------------+------------+--------------+-----------------+--------------+-----------------+---------------+
|      P1      |     8      |       0      |        22       |      14      |      22         |       0       |
+--------------+------------+--------------+-----------------+--------------+-----------------+---------------+
|      P2      |     2      |       5      |        11       |       4      |       6         |       4       |
+--------------+------------+--------------+-----------------+--------------+-----------------+---------------+
|      P3      |     7      |       1      |        23       |      15      |      22         |       2       |
+--------------+------------+--------------+-----------------+--------------+-----------------+---------------+
|      P4      |     3      |       6      |        14       |       5      |       8         |       5       |
+--------------+------------+--------------+-----------------+--------------+-----------------+---------------+
|      P5      |     5      |       8      |        25       |      12      |      17         |       9       |
+--------------+------------+--------------+-----------------+--------------+-----------------+---------------+


Average Completion time : 19.00ns
Average Waiting time : 10.00ns
Average TurnAround time : 15.00ns
Average Response time : 4.00ns
```