Problem Solution Code

```
#include<stdio.h>
#include<iostream>
#include<stdlib.h>
    int size,arival_time[100],bust_time[100],completion_time[100],waitin
g_time[100],trunaround_time[100],remanning_bt[100],total_time_quantam=0,
read_queue[100];
    double avg_trunaround_time=0,avg_waiting_time=0;
    int current_time,Remening_proc=size,indicator,time_quantam_i1,time_q
uantam_i2,time_quantam_itration,remmaning_process;
int main()
{
    using namespace std;
    cout<<"Enter the number of process :";
    cin>>size;
    cout<<"\n\nEnter the arrival time and burst time of the processes\n"
;
    for(int process_no=0;process_no<size;process_no++)
    {
        cout<<"\nProcess P"<<process_no+1<<"\n";
        cout<<"\tArrival time = ";
        cin>>arival_time[process_no];
        cout<<"\tBurst time = ";
        cin>>bust_time[process_no];
        remanning_bt[process_no]=bust_time[process_no];
        total_time_quantam+=bust_time[process_no];
    }
    system("CLS");
    cout<<"The details of time quantum are as follows:\n";
    cout<<"\nThe time quantum for first Itration is 3.\n";
    cout<<"The time quantum for first Itration is 6.\n";
    cout<<"After second itration the Shortest job will assign CPU.\n\n";
    time_quantam_i1=3;
    time_quantam_i2=6;
    time_quantam_itration=1;
    current_time=0;
    remmaning_process=size;
    for(int Process_no=0;Process_no<remmaning_process;Process_no++)
    {
        if(remanning_bt[Process_no]<time_quantam_i1 && remanning_bt[Proc
ess_no]>=0&&current_time<9)
        {
            current_time+=remanning_bt[Process_no];
            remanning_bt[Process_no]=0;
```

```
            indicator = 1;
            time_quantam_itration++;
            remmaning_process--;
        }
        else if(remanning_bt[Process_no]>0&&current_time<9)
        {
            if(time_quantam_itration==1)
            {remanning_bt[Process_no]-=time_quantam_i1;
            time_quantam_itration++;
            current_time+=time_quantam_i1;}
            else if(time_quantam_itration==2)
            {remanning_bt[Process_no]-=time_quantam_i2;
            current_time+=time_quantam_i2;}
        }
        else if(remanning_bt[Process_no]<9 && remanning_bt[Process_no]>=
3&&current_time<9)
        {
            current_time+=remanning_bt[Process_no];
            remanning_bt[Process_no]=0;
            remmaning_process--;
            indicator = 1;
        }
        else if(remanning_bt[Process_no]>3&&current_time<9)
        {
            remanning_bt[Process_no]-=time_quantam_i2;
            current_time+=time_quantam_i2;
        }
        if(remanning_bt[Process_no]==0 && indicator==1)
        {
            Remening_proc--;
            completion_time[Process_no]=current_time;
            cout<<completion_time[Process_no];
            trunaround_time[Process_no]=completion_time[Process_no]-
arival_time[Process_no];
            waiting_time[Process_no]=trunaround_time[Process_no]-
bust_time[Process_no];
            indicator = 0;

        }

    }

    for(int Process_no=0;Process_no<remmaning_process;Process_no++)
    {
        int min =remanning_bt[0];
```

```
        int i = 0,j=0;
    for (i; i < size; i++)
    {
        if (min > remanning_bt[i] && current_time>arival_time[i])
            {
                min = remanning_bt[i];
            }
    }

    for (j; j < size; j++)
    {
        if(remanning_bt[j]==min)
        break;
   }
        if(current_time>arival_time[j] && min!=100000)
        {
            remanning_bt[j]=100000;
            current_time+=min;
            completion_time[j]=current_time;
            trunaround_time[j]=completion_time[j]-arival_time[j];
            waiting_time[j]=trunaround_time[j]-bust_time[j];
        }
    }


    cout<<"\nProcess\t\tArival time\tBurst time\tComplection time\tTurna
round Time\t\twaiting time";
    for(int i=0;i<size;i++)
    {
        cout<<"\nP"<<i+1<<"\t\t"<<arival_time[i]<<"\t\t"<<bust_time[i]<<
"\t\t"<<completion_time[i]<<"\t\t\t"<<trunaround_time[i]<<"\t\t\t"<<wait
ing_time[i];
    }
    for(int k=0;k<size;k++)
    {
        avg_waiting_time+=waiting_time[k];
        avg_trunaround_time+=trunaround_time[k];
    }
    cout<<"\n\n Average Trunaround time : "<<(avg_trunaround_time)/size<
<endl;
    cout<<" Average Waiting time : "<<(avg_waiting_time)/size<<endl;
}
```