Implementation code

```
#include <bits/stdc++.h>
using namespace std;
int n;
vector<int> p(n), bt(n);
void input_view()
{
    cout << "Process   " << "Burst_Time(BT)" << endl;
    for (int i = 0; i < n; i++)
    {
        cout << "   P" << p[i] << " \t\t" << bt[i] << endl;
    }
}
int main()
{
    cout << "Enter the number of processes: ";
    cin >> n;

    cout << "Enter Burst Time(BT) for each processes: ";
    for (int i = 0; i < n; i++)
    {
        int x;
        cin >> x;
        bt.push_back(x);
        p.push_back(i + 1);
    }
    // input view
    cout << endl
         << "Input view:" << endl;
    input_view();

    // sorting according to Burst time
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (bt[i] < bt[j])
            {
                swap(bt[i], bt[j]);
```

```
                swap(p[i], p[j]);
            }
        }
    }
    // view of inputs after sorting
    cout << endl
         << "After sorting, View of inputs:" << endl;
    input_view();

    // calculation of CT,TAT,WT,RT
    vector<int> ct(n), tat(n), wt(n), rt(n);
    int sum = 0;
    int r = 0;
    for (int i = 0; i < n; i++)
    {
        sum += bt[i];
        ct[i] = sum;
        tat[i] = ct[i];
        wt[i] = tat[i] - bt[i];
        rt[i] = r;
        r += bt[i];
    }
    // Average waiting time
    float avg_wt, s = 0;
    for (int i = 0; i < n; i++)
    {
        s += wt[i];
    }
    avg_wt = s / n;
    // Gantt chart
    cout << endl
         << "\t Gantt Chart : " << endl;
    for (int i = 0; i < n; i++)
    {
        cout << "--------";
    }
    cout << endl;

    for (int i = 0; i < n; i++)
    {
```

```
        cout << "|  P" << p[i] << "   ";
    }

    cout << "|" << endl;

    for (int i = 0; i < n; i++)
    {
        cout << "--------";
    }
    cout << endl;

    cout << 0;
    for (int i = 0; i < n; i++)
    {
        cout << "\t" << ct[i];
    }
    cout << endl;

    // view of outputs
    cout << endl
         << "\t Final Result: " << endl;
    cout << "Process   " << "Burst_Time(BT)   " << "Completion_Time(CT)
" << "Turnaround_Time(TAT)   " << "Waiting_Time(WT)   " <<
"Response_Time(RT)" << endl;
    for (int i = 0; i < n; i++)
    {
        cout << "   P" << p[i] << " \t\t" << bt[i] << " \t\t" << ct[i] <<
" \t\t\t" << tat[i] << " \t\t\t" << wt[i] << " \t\t" << rt[i] << endl;
    }

    // Average waiting time
    cout << endl
         << "Average Waiting Time: " << avg_wt << endl;
    return 0;
}
```

Result Analysis

```
Enter the number of processes: 4
Enter Burst Time(BT) for each processes: 21 3 6 2

Input view:
Process   Burst_Time(BT)
   P1           21
   P2           3
   P3           6
   P4           2

After sorting, View of inputs:
Process   Burst_Time(BT)
   P4           2
   P2           3
   P3           6
   P1           21

         Gantt Chart :
-------------------------------
|  P4   |  P2   |  P3   |  P1   |
-------------------------------
0       2       5       11      32

         Final Result:
Process   Burst_Time(BT)   Completion_Time(CT)   Turnaround_Time(TAT)   Waiting_Time(WT)   Response_Time(RT)
   P4           2               2                      2                      0                  0
   P2           3               5                      5                      2                  2
   P3           6               11                     11                     5                  5
   P1           21              32                     32                     11                 11

Average Waiting Time: 4.5
```