Implementation code:

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

    cout << "Enter Arrival Time(AT) for " << n << " processes: ";
    for (int i = 0; i < n; i++)
    {
        int x;
        cin >> x;
        at.push_back(x);
        p.push_back(i + 1);
    }

    cout << "Enter Burst Time(BT) for " << n << " processes: ";
    for (int i = 0; i < n; i++)
    {
        int x;
        cin >> x;
        bt.push_back(x);
    }

    // inputs visible
    cout << endl
         << "\t View of inputs:" << endl;
    input_view();

    // sorting process according to AT.
    for (int i = 0; i < n; i++)
```

```
    {
        for (int j = 0; j < n; j++)
        {
            if (at[j] > at[j+1])
            {
                swap(p[i], p[j]);
                swap(at[i], at[j]);
                swap(bt[i], bt[j]);
            }
        }
    }

    // view of inputs after sorting
    cout << endl
         << "\t After sorting, View of inputs:" << endl;
    input_view();

    vector<int> ct(n), tat(n), wt(n), rt(n);

    // calculation of CT,TAT,WT,RT
    int sum = at[0];
    rt[0] = at[0];
    for (int i = 0; i < n; i++)
    {
        ct[i] = sum + bt[i];
        sum = ct[i];

        tat[i] = ct[i] - at[i];

        wt[i] = tat[i] - bt[i];

        rt[i + 1] = ct[i] - at[i + 1];
    }

    // Grand Chart
    cout << endl
         << "\t Grand Chart: " << endl
         << endl;
    for (int i = 0; i < n; i++)
    {
        cout << "|   P" << p[i] << "  ";
    }
    cout << " |" << endl;
    cout << at[0];
    for (int i = 0; i < n; i++)
```

```
    {
        cout << "\t" << ct[i];
    }
    // formating according to processes like(p1,p2,p3...)
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (p[i] < p[j])
            {
                swap(p[i], p[j]);
                swap(at[i], at[j]);
                swap(bt[i], bt[j]);
                swap(ct[i], ct[j]);
                swap(tat[i], tat[j]);
                swap(wt[i], wt[j]);
                swap(rt[i], rt[j]);
            }
        }
    }
    // Final result
    cout << endl
         << endl
         << "\t Final result:" << endl
         << endl;
    cout << "Pro.\t" << "AT\t" << "BT\t" << "CT\t" << "TAT\t" << "WT\t" << "RT\t"
<< endl;
    for (int i = 0; i < n; i++)
    {
        cout << "P" << p[i] << "\t" << at[i] << "\t" << bt[i] << "\t" << ct[i]
             << "\t" << tat[i] << "\t" << wt[i] << "\t" << rt[i] << endl;
    }
    return 0;
}
```

Result Analysis:

```
Enter the number of process: 3
Enter Arrival Time(AT) for 3 times: 2 0 4
Enter Burst Time(BT) for 3 times: 5 3 4

         View of inputs:
Process   Arrival_Time(AT)   Burst_Time(BT)
   P1           2                5
   P2           0                3
   P3           4                4

         After sorting, View of inputs:
Process   Arrival_Time(AT)   Burst_Time(BT)
   P2           0                3
   P1           2                5
   P3           4                4

         Grand Chart:

|   P2  |   P1  |   P3   |
0       3       8       12

         Final result:

Pro.    AT      BT      CT      TAT     WT      RT
P1      2       5       8       6       1       1
P2      0       3       3       3       0       0
P3      4       4       12      8       4       4
```