

Source code:

```
// Adit Luhadia - 190911112
#include <iostream>
#include <algorithm>
using namespace std;

struct node
{
    char pname;
    int btime;
    int atime;
    int priority;
    int restime = 0;
    int ctime = 0;
    int wtime = 0;
} a[1000], b[1000], c[1000];

void insert(int n)
{
    int i;
    for (i = 0; i < n; i++)
    {
        a[i].pname = (char)(i + 49);
        cin >> a[i].atime;
        cin >> a[i].btime;
        cin >> a[i].priority;
        a[i].wtime = -a[i].atime + 1;
    }
}

bool btimeSort(node a, node b)
{
    return a.btime < b.btime;
}

bool atimeSort(node a, node b)
{
    return a.atime < b.atime;
}

bool prioritySort(node a, node b)
{
    return a.priority < b.priority;
}

int k = 0, f = 0, r = 0;

void disp(int nop, int qt)
```

```

{
    int n = nop, q;
    sort(a, a + n, atimeSort);
    int ttime = 0, i;
    int j, tArray[n];
    int alltime = 0;
    bool moveLast = false;
    for (i = 0; i < n; i++)
    {
        alltime += a[i].btime;
    }
    alltime += a[0].atime;
    for (i = 0; ttime <= alltime;)
    {
        j = i;
        while (a[j].atime <= ttime && j != n)
        {
            b[r] = a[j];
            j++;
            r++;
        }
        if (r == f)
        {
            c[k].pname = 'i';
            c[k].btime = a[j].atime - ttime;
            c[k].atime = ttime;
            ttime += c[k].btime;
            k++;
            continue;
        }
        i = j;
        if (moveLast == true)
        {
            sort(b + f, b + r, prioritySort);
        }

        j = f;
        if (b[j].btime > qt)
        {
            c[k] = b[j];
            c[k].btime = qt;
            k++;
            b[j].btime = b[j].btime - qt;
            ttime += qt;
            moveLast = true;
            for (q = 0; q < n; q++)
            {
                if (b[j].pname != a[q].pname)

```

```

        {
            a[q].wtime += qt;
        }
    }
}
else
{
    c[k] = b[j];
    k++;
    f++;
    ttime += b[j].btime;
    moveLast = false;
    for (q = 0; q < n; q++)
    {
        if (b[j].pname != a[q].pname)
        {
            a[q].wtime += b[j].btime;
        }
    }
}
if (f == r && i >= n)
    break;
}
tArray[i] = ttime;
ttime += a[i].btime;
for (i = 0; i < k - 1; i++)
{
    if (c[i].pname == c[i + 1].pname)
    {
        c[i].btime += c[i + 1].btime;
        for (j = i + 1; j < k - 1; j++)
            c[j] = c[j + 1];
        k--;
        i--;
    }
}

int rtime = 0;
for (j = 0; j < n; j++)
{
    rtime = 0;
    for (i = 0; i < k; i++)
    {
        if (c[i].pname == a[j].pname)
        {
            a[j].restime = rtime;
            break;
        }
    }
}

```

```

        rtime += c[i].btime;
    }
}

float averageWaitingTime = 0;
float averageResponseTime = 0;
float averageTAT = 0;

cout << "\nGantt Chart\n";
rtime = 0;
for (i = 0; i < k; i++)
{
    if (i != k)
        cout << "| " << 'P' << c[i].pname << " ";
    rtime += c[i].btime;
    for (j = 0; j < n; j++)
    {
        if (a[j].pname == c[i].pname)
            a[j].ctime = rtime;
    }
}
cout << "\n";
rtime = 0;
for (i = 0; i < k + 1; i++)
{
    cout << rtime << "\t";
    tArray[i] = rtime;
    rtime += c[i].btime;
}

cout << "\n";
cout << "\n";
cout << "P.Name Priority AT\tBT\tCT\tTAT\tWT\tRT\n";
for (i = 0; i < nop && a[i].pname != 'i'; i++)
{
    if (a[i].pname == '\0')
        break;
    cout << 'P' << a[i].pname << "\t";
    cout << a[i].priority << "\t";
    cout << a[i].atime << "\t";
    cout << a[i].btime << "\t";
    cout << a[i].ctime << "\t";
    cout << a[i].wtime + a[i].ctime - rtime + a[i].btime << "\t";
    averageTAT += a[i].wtime + a[i].ctime - rtime + a[i].btime;
    cout << a[i].wtime + a[i].ctime - rtime << "\t";
    averageWaitingTime += a[i].wtime + a[i].ctime - rtime;
    cout << a[i].restime - a[i].atime << "\t";
    averageResponseTime += a[i].restime - a[i].atime;
}

```

```

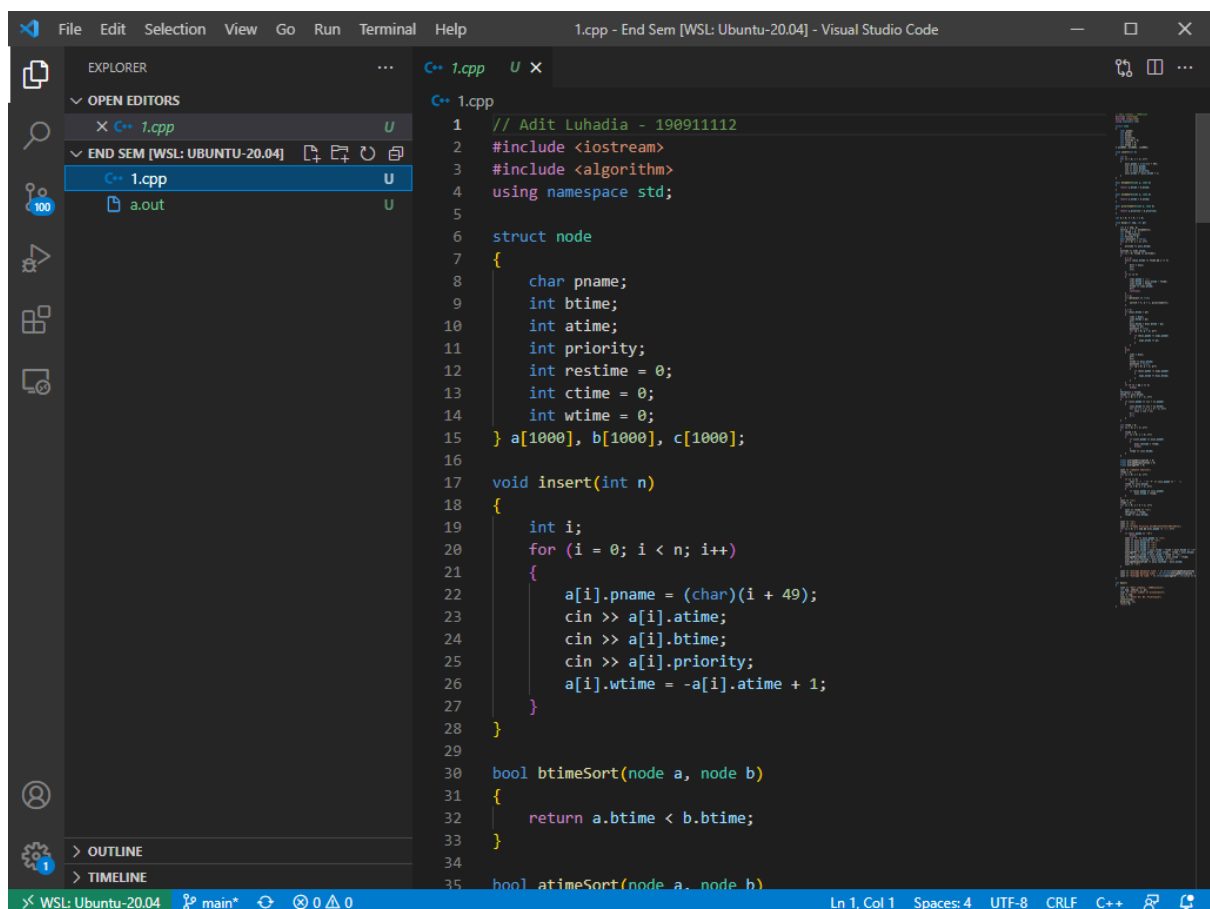
        cout << "\n";
    }

    cout << "Average Response time: " << (float)averageResponseTime / (float)n
    << endl;
    cout << "Average Waiting time: " << (float)averageWaitingTime / (float)n <
    < endl;
    cout << "Average TA time: " << (float)averageTAT / (float)n << endl;
}

int main()
{
    cout << "Adit Luhadia - 190911112\n";
    int nop, choice, i, qt;
    cout << "Enter number of processes\n";
    cin >> nop;
    cout << "Enter AT, BT, Priority\n";
    insert(nop);
    disp(nop, 1);
    return 0;
}

```

Program screenshots:



```
33 }
34
35 bool atimeSort(node a, node b)
36 {
37     return a.atime < b.atime;
38 }
39
40 bool prioritySort(node a, node b)
41 {
42     return a.priority < b.priority;
43 }
44
45 int k = 0, f = 0, r = 0;
46
47 void disp(int nop, int qt)
48 {
49     int n = nop, q;
50     sort(a, a + n, atimeSort);
51     int ttime = 0, i;
52     int j, tArray[n];
53     int alltime = 0;
54     bool moveLast = false;
55     for (i = 0; i < n; i++)
56     {
57         alltime += a[i].btime;
58     }
59     alltime += a[0].atime;
60     for (i = 0; ttime <= alltime;)
61     {
62         j = i;
63         while (a[j].atime <= ttime && j != n)
64         {
65             b[r] = a[j];
66             j++;
```

```
67         r++;
68     }
69     if (r == f)
70     {
71         c[k].pname = 'i';
72         c[k].btime = a[j].atime - ttime;
73         c[k].atime = ttime;
74         ttime += c[k].btime;
75         k++;
76         continue;
77     }
78     i = j;
79     if (moveLast == true)
80     {
81         sort(b + f, b + r, prioritySort);
82     }
83
84     j = f;
85     if (b[j].btime > qt)
86     {
87         c[k] = b[j];
88         c[k].btime = qt;
89         k++;
90         b[j].btime = b[j].btime - qt;
91         ttime += qt;
92         moveLast = true;
93         for (q = 0; q < n; q++)
94         {
95             if (b[j].pname != a[q].pname)
96             {
97                 a[q].wtime += qt;
98             }
99         }
100     }
101 }
```

```
101     else
102     {
103         c[k] = b[j];
104         k++;
105         f++;
106         ttime += b[j].btime;
107         moveLast = false;
108         for (q = 0; q < n; q++)
109         {
110             if (b[j].pname != a[q].pname)
111             {
112                 a[q].wtime += b[j].btime;
113             }
114         }
115     }
116     if (f == r && i >= n)
117         break;
118 }
119 tArray[i] = ttime;
120 ttime += a[i].btime;
121 for (i = 0; i < k - 1; i++)
122 {
123     if (c[i].pname == c[i + 1].pname)
124     {
125         c[i].btime += c[i + 1].btime;
126         for (j = i + 1; j < k - 1; j++)
127             c[j] = c[j + 1];
128         k--;
129         i--;
130     }
131 }
132
133 int rtime = 0;
134 for (j = 0; j < n; j++)
135 {
```

```
135 {
136     rtime = 0;
137     for (i = 0; i < k; i++)
138     {
139         if (c[i].pname == a[j].pname)
140         {
141             a[j].restime = rtime;
142             break;
143         }
144         rtime += c[i].btime;
145     }
146 }
147
148 float averageWaitingTime = 0;
149 float averageResponseTime = 0;
150 float averageTAT = 0;
151
152 cout << "\nGantt Chart\n";
153 rtime = 0;
154 for (i = 0; i < k; i++)
155 {
156     if (i != k)
157         cout << " | " << 'P' << c[i].pname << " ";
158     rtime += c[i].btime;
159     for (j = 0; j < n; j++)
160     {
161         if (a[j].pname == c[i].pname)
162             a[j].ctime = rtime;
163     }
164 }
165 cout << "\n";
166 rtime = 0;
167 for (i = 0; i < k + 1; i++)
168 {
169     cout << rtime << "\t";
```

This screenshot shows the Visual Studio Code editor with a C++ file named `1.cpp` open. The Explorer sidebar on the left shows the file structure with `1.cpp` and `a.out` under the `END SEM [WSL: UBUNTU-20.04]` folder. The main editor area displays the following C++ code:

```
167     for (i = 0; i < k + 1; i++)
168     {
169         cout << rtime << "\t";
170         tArray[i] = rtime;
171         rtime += c[i].btime;
172     }
173
174     cout << "\n";
175     cout << "\n";
176     cout << "P.Name Priority AT\BT\CT\TAT\TWT\TRT\n";
177     for (i = 0; i < nop && a[i].pname != 'i'; i++)
178     {
179         if (a[i].pname == '\0')
180             break;
181         cout << 'P' << a[i].pname << "\t";
182         cout << a[i].priority << "\t";
183         cout << a[i].atime << "\t";
184         cout << a[i].btime << "\t";
185         cout << a[i].ctime << "\t";
186         cout << a[i].wtime + a[i].ctime - rtime + a[i].btime << "\t";
187         averageTAT += a[i].wtime + a[i].ctime - rtime + a[i].btime;
188         cout << a[i].wtime + a[i].ctime - rtime << "\t";
189         averageWaitingTime += a[i].wtime + a[i].ctime - rtime;
190         cout << a[i].restime - a[i].atime << "\t";
191         averageResponseTime += a[i].restime - a[i].atime;
192         cout << "\n";
193     }
194
195     cout << "Average Response time: " << (float)averageResponseTime
196     cout << "Average Waiting time: " << (float)averageWaitingTime /
197     cout << "Average TA time: " << (float)averageTAT / (float)n << endl;
198 }
199
200 int main()
201 {
```

The status bar at the bottom indicates the file is in WSL: Ubuntu-20.04, the main function is active, and the editor is in UTF-8 encoding with CRLF line endings.

This screenshot shows the Visual Studio Code editor with the same C++ file `1.cpp` open. The Explorer sidebar shows the file structure. The main editor area displays the following C++ code:

```
198 }
199
200 int main()
201 {
202     cout << "Adit Luhadia - 190911112\n";
203     int nop, choice, i, qt;
204     cout << "Enter number of processes\n";
205     cin >> nop;
206     cout << "Enter AT, BT, Priority\n";
207     insert(nop);
208     disp(nop, 1);
209     return 0;
210 }
```

The status bar at the bottom indicates the file is in WSL: Ubuntu-20.04, the main function is active, and the editor is in UTF-8 encoding with CRLF line endings.

Output screenshots:

Sample run given in the question:

```
masteradit@LAPTOP-UIIM4O9R x + v - □ x
masteradit@LAPTOP-UIIM4O9RP:/mnt/c/Users/aditl/Documents/IV-Sem-Labs/Operating Systems Lab/End Sem$ g++ 1.cpp
masteradit@LAPTOP-UIIM4O9RP:/mnt/c/Users/aditl/Documents/IV-Sem-Labs/Operating Systems Lab/End Sem$ ./a.out
Adit Luhadia - 190911112
Enter number of processes
4
Enter AT, BT, Priority
0 4 4
1 2 2
2 3 3
3 2 1

Gantt Chart
| P1 | P2 | P3 | P4 | P3 | P1 |
0 1 3 4 6 8 11

P.Name Priority AT BT CT TAT WT RT
P1 4 0 4 11 11 7 0
P2 2 1 2 3 2 0 0
P3 3 2 3 8 6 3 1
P4 1 3 2 6 3 1 1
Average Response time: 0.5
Average Waiting time: 2.75
Average TA time: 5.5
masteradit@LAPTOP-UIIM4O9RP:/mnt/c/Users/aditl/Documents/IV-Sem-Labs/Operating Systems Lab/End Sem$
```

I tried a few more sample runs to test the program:

```
masteradit@LAPTOP-UIIM4O9R x + v - □ x
masteradit@LAPTOP-UIIM4O9RP:/mnt/c/Users/aditl/Documents/IV-Sem-Labs/Operating Systems Lab/End Sem$ g++ 1.cpp
masteradit@LAPTOP-UIIM4O9RP:/mnt/c/Users/aditl/Documents/IV-Sem-Labs/Operating Systems Lab/End Sem$ ./a.out
Adit Luhadia - 190911112
Enter number of processes
5
Enter AT, BT, Priority
0 3 2
1 2 4
2 5 3
4 2 1
3 3 5

Gantt Chart
| P1 | P3 | P4 | P3 | P2 | P5 |
0 3 4 6 10 12 15

P.Name Priority AT BT CT TAT WT RT
P1 2 0 3 3 3 0 0
P2 4 1 2 12 11 9 9
P3 3 2 5 10 8 3 1
P5 5 3 3 15 12 9 9
P4 1 4 2 6 2 0 0
Average Response time: 3.8
Average Waiting time: 4.2
Average TA time: 7.2
masteradit@LAPTOP-UIIM4O9RP:/mnt/c/Users/aditl/Documents/IV-Sem-Labs/Operating Systems Lab/End Sem$ |
```

```
masteradit@LAPTOP-UIIM4O9R x + v - □ x
masteradit@LAPTOP-UIIM4O9RP:/mnt/c/Users/aditl/Documents/IV-Sem-Labs/Operating Systems Lab/End Sem$ ./a.out
Adit Luhadia - 190911112
Enter number of processes
3
Enter AT, BT, Priority
0 2 2
1 4 3
2 5 1

Gantt Chart
| P1 | | P2 | | P3 | | P2 |
0    2    3    8    11

P.Name Priority AT    BT    CT    TAT    WT    RT
P1      2      0      2      2      2      0      0
P2      3      1      4     11     10      6      1
P3      1      2      5      8      6      1      1
Average Response time: 0.666667
Average Waiting time: 2.33333
Average TA time: 6
masteradit@LAPTOP-UIIM4O9RP:/mnt/c/Users/aditl/Documents/IV-Sem-Labs/Operating Systems Lab/End Sem$ |
```