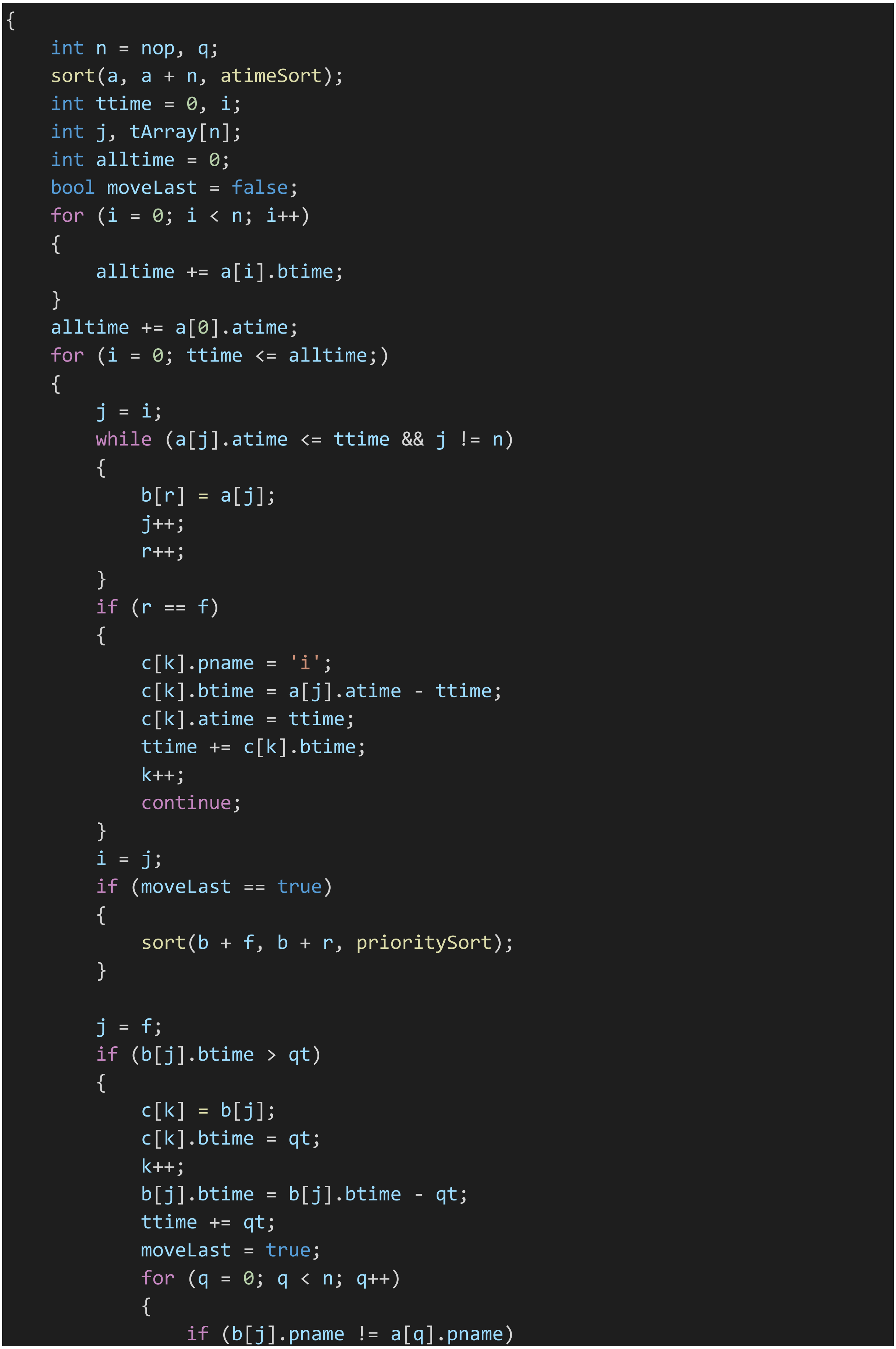
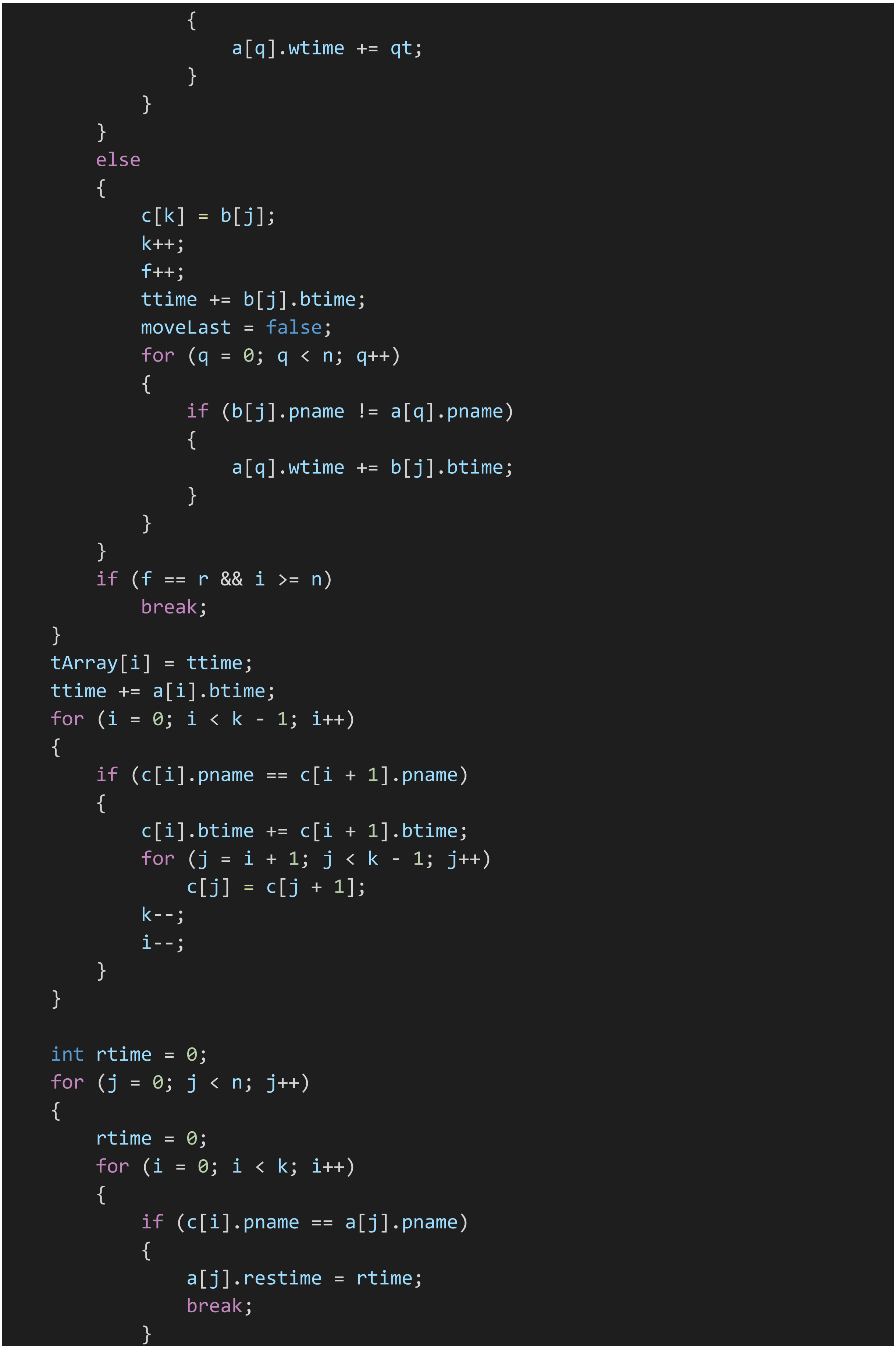
Source code:

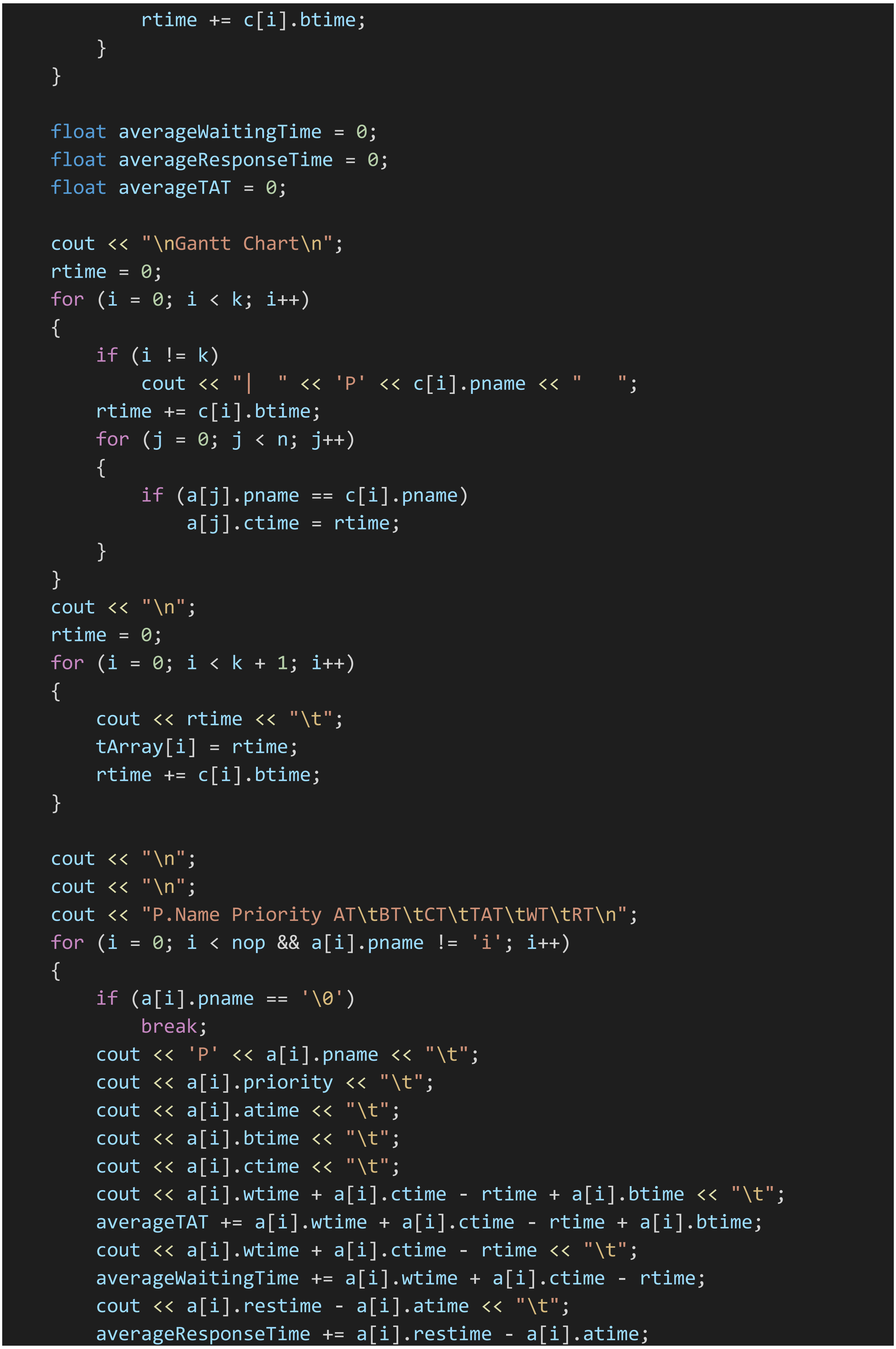
```
// Adit Luhadia - 190911112
#include <iostream>
#include <algorithm>
using namespace std;

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)
```

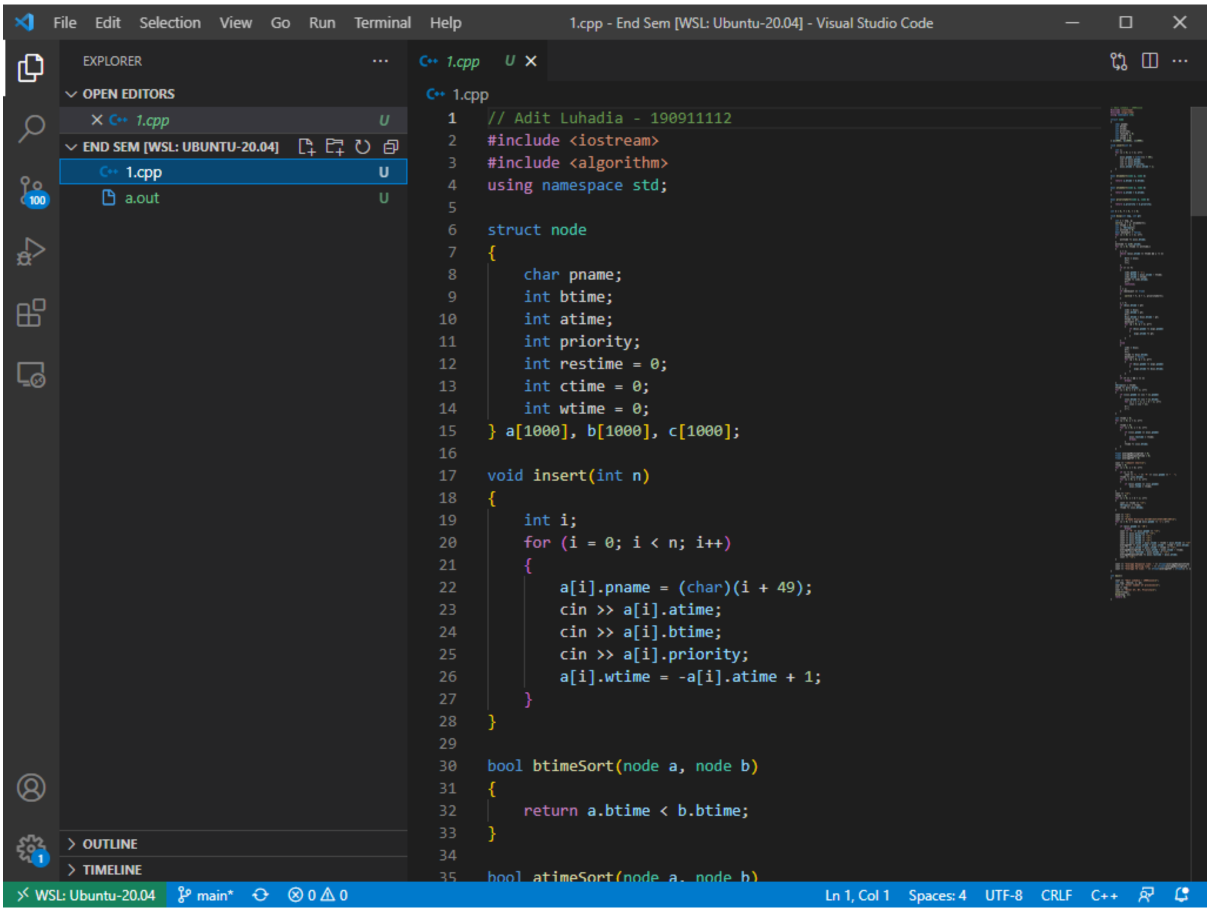


```
            {
                a[q].wtime += qt;
            }
        }
    }
    else
    {
        c[k] = b[j];
        k++;
        f++;
        ttime += b[j].btime;
        moveLast = false;
        for (q = 0; q < n; q++)
        {
            if (b[j].pname != a[q].pname)
            {
                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;
        }
```

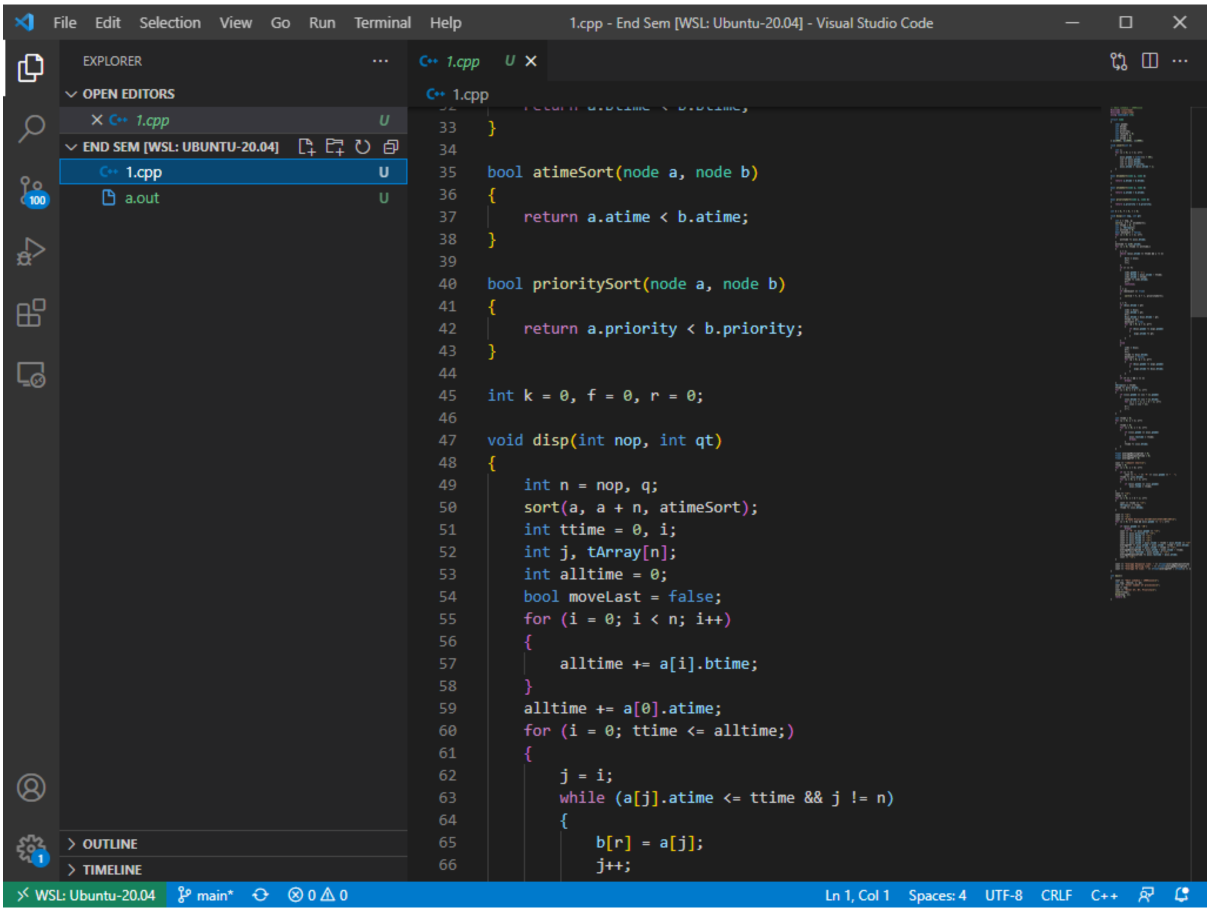


```
            rtime += c[i].btime;
        }
    }

    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;
}
```

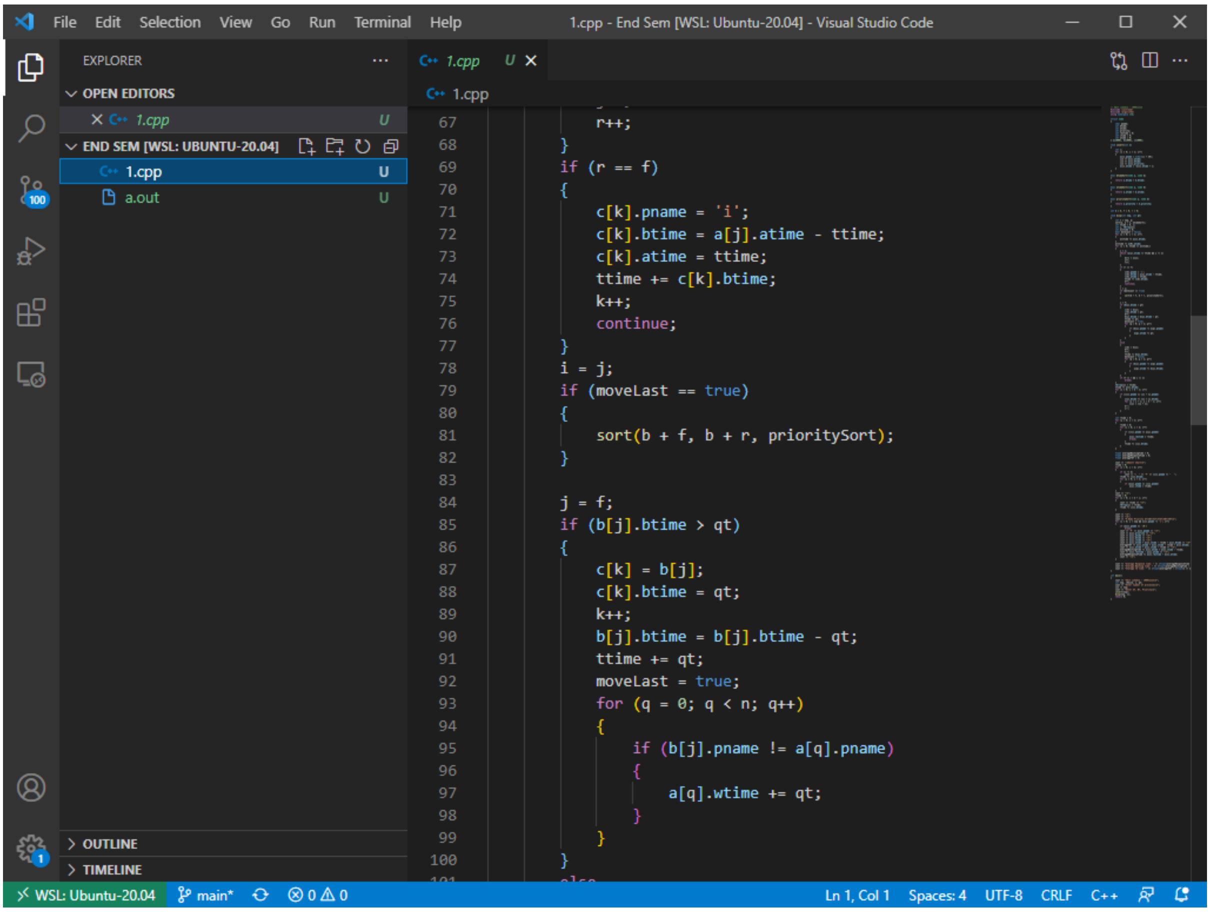
Program screenshots:


File Edit Selection View Go Run Terminal Help
1.cpp - End Sem [WSL: Ubuntu-20.04] - Visual Studio Code
EXPLORER
OPEN EDITORS
1.cpp U
END SEM [WSL: UBUNTU-20.04]
1.cpp U
a.out U
OUTLINE
TIMELINE
1.cpp U
1.cpp
WSL: Ubuntu-20.04
main*
0 ⚠ 0
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF C++

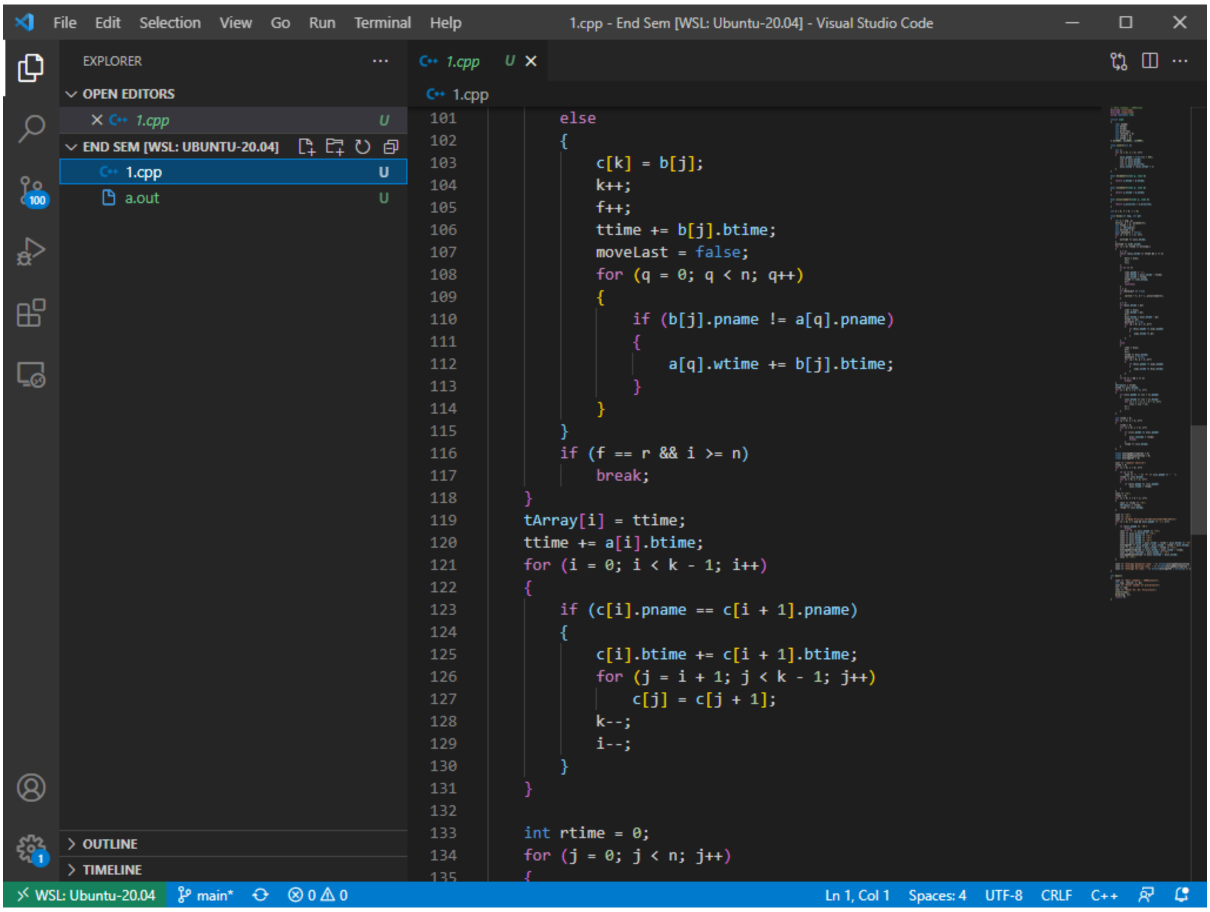
```
// Adit Luhadia - 190911112
#include <iostream>
#include <algorithm>
using namespace std;

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)
```


File Edit Selection View Go Run Terminal Help
1.cpp - End Sem [WSL: Ubuntu-20.04] - Visual Studio Code
EXPLORER
OPEN EDITORS
1.cpp U
END SEM [WSL: UBUNTU-20.04]
1.cpp U
a.out U
OUTLINE
TIMELINE
WSL: Ubuntu-20.04
main*
0 0
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF C++

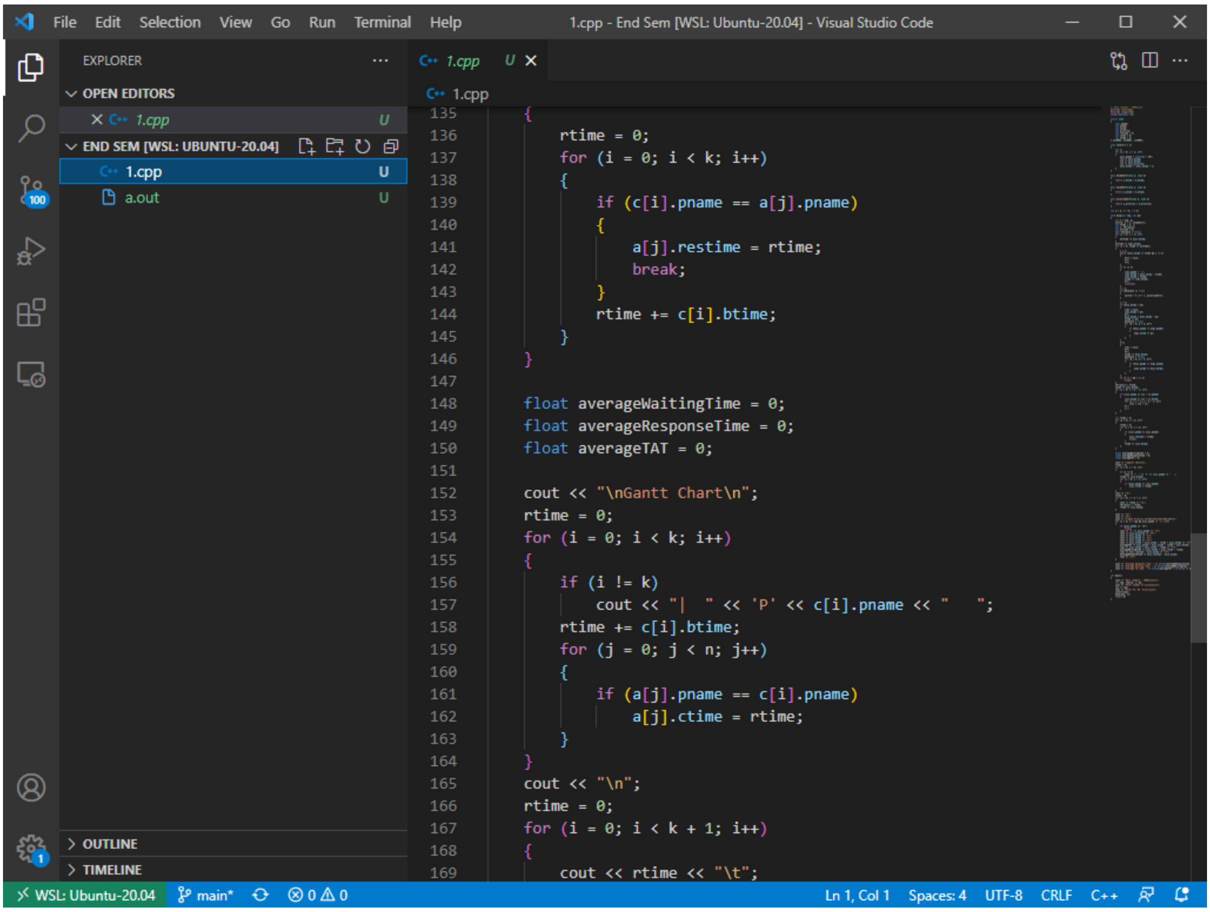
```
    return a.btime < b.btime;
}

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++;
```


File Edit Selection View Go Run Terminal Help
1.cpp - End Sem [WSL: Ubuntu-20.04] - Visual Studio Code
EXPLORER
OPEN EDITORS
1.cpp U
END SEM [WSL: UBUNTU-20.04]
1.cpp U
a.out U
OUTLINE
TIMELINE
WSL: Ubuntu-20.04
main*
0 0
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF C++

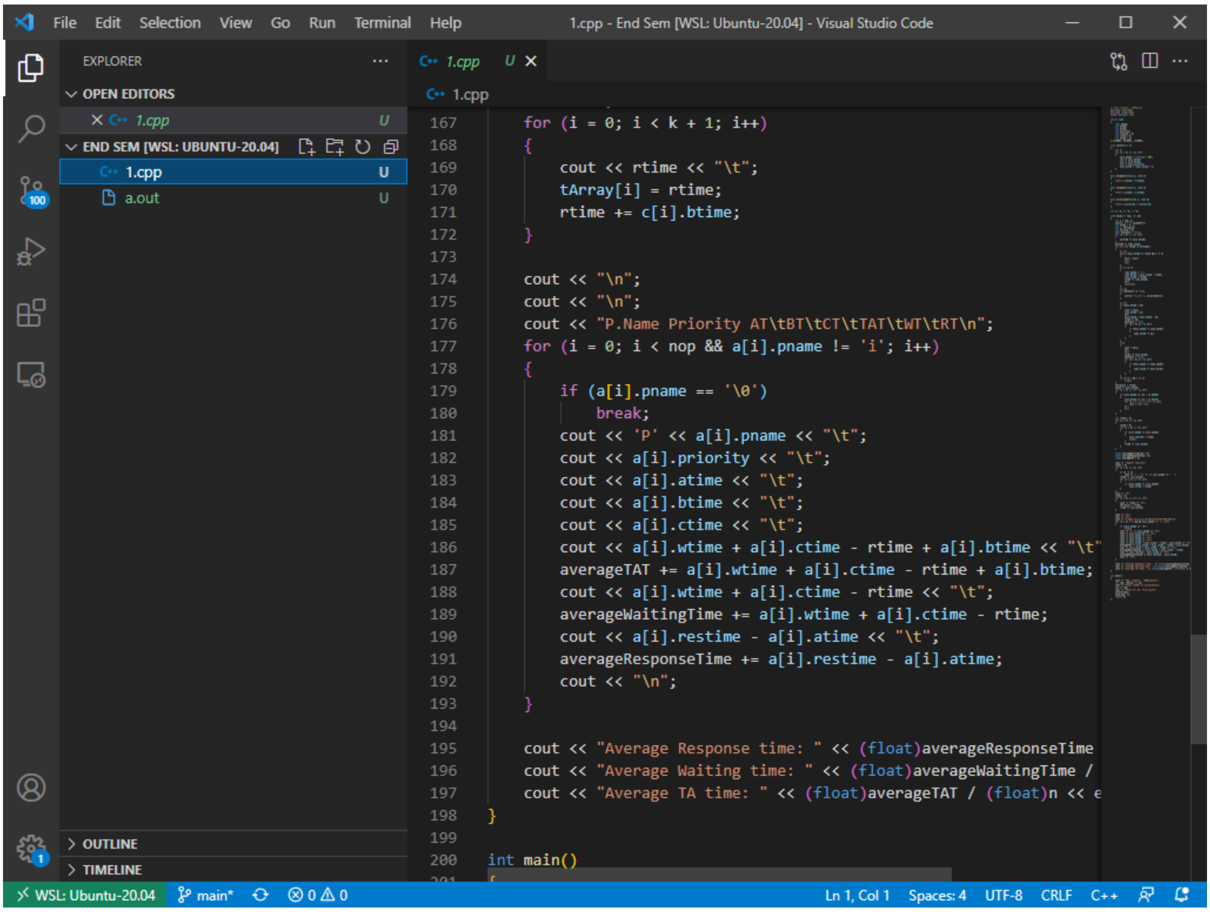
```
            r++;
        }
        if (r == f)
        {
            c[k].pname = 'i';
            c[k].btime = a[j].atime - ttime;
            c[k].atime = ttime;
            ttime += c[k].btime;
            k++;
            continue;
        }
        i = j;
        if (moveLast == true)
        {
            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
```


File Edit Selection View Go Run Terminal Help
1.cpp - End Sem [WSL: Ubuntu-20.04] - Visual Studio Code
EXPLORER
OPEN EDITORS
1.cpp U
END SEM [WSL: UBUNTU-20.04]
1.cpp U
a.out U
OUTLINE
TIMELINE
1.cpp
WSL: Ubuntu-20.04
main*
0 0
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF C++

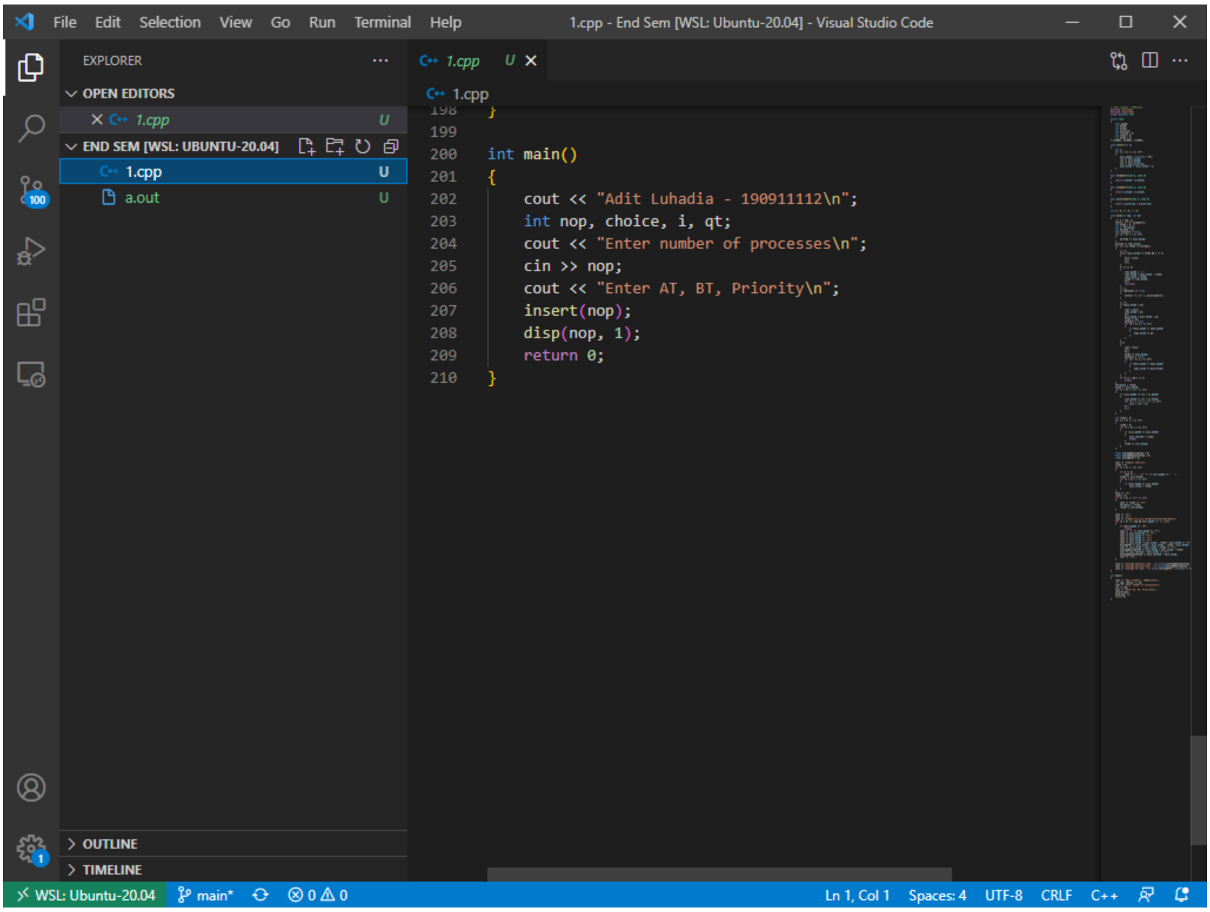
```
            else
            {
                c[k] = b[j];
                k++;
                f++;
                ttime += b[j].btime;
                moveLast = false;
                for (q = 0; q < n; q++)
                {
                    if (b[j].pname != a[q].pname)
                    {
                        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++)
```


File Edit Selection View Go Run Terminal Help
1.cpp - End Sem [WSL: Ubuntu-20.04] - Visual Studio Code
EXPLORER
OPEN EDITORS
1.cpp U
END SEM [WSL: UBUNTU-20.04]
1.cpp U
a.out U
OUTLINE
TIMELINE
1.cpp
WSL: Ubuntu-20.04
main*
0 0
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF C++

```
        {
            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";
```


File Edit Selection View Go Run Terminal Help
1.cpp - End Sem [WSL: Ubuntu-20.04] - Visual Studio Code
EXPLORER
OPEN EDITORS
1.cpp U
END SEM [WSL: UBUNTU-20.04]
1.cpp U
a.out U
OUTLINE
TIMELINE
WSL: Ubuntu-20.04
main*
0 0
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF C++

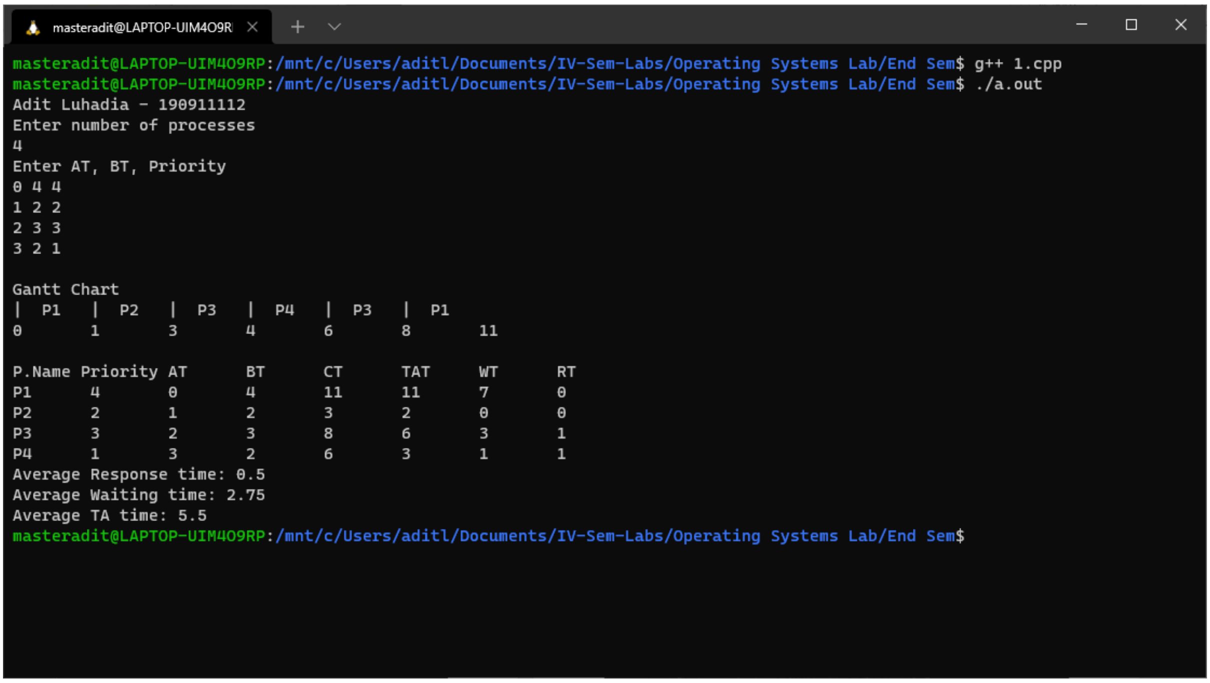
```cpp
    for (i = 0; i < k + 1; i++)
    {
        cout << rtime << "\t";
        tArray[i] = rtime;
        rtime += c[i].btime;
    }

    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
    cout << "Average Waiting time: " << (float)averageWaitingTime /
    cout << "Average TA time: " << (float)averageTAT / (float)n << e
}

int main()
```


File Edit Selection View Go Run Terminal Help
1.cpp - End Sem [WSL: Ubuntu-20.04] - Visual Studio Code
EXPLORER
OPEN EDITORS
1.cpp U
END SEM [WSL: UBUNTU-20.04]
1.cpp U
a.out U
OUTLINE
TIMELINE
WSL: Ubuntu-20.04
main*
0 0
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF C++

```cpp
}

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;
}
```

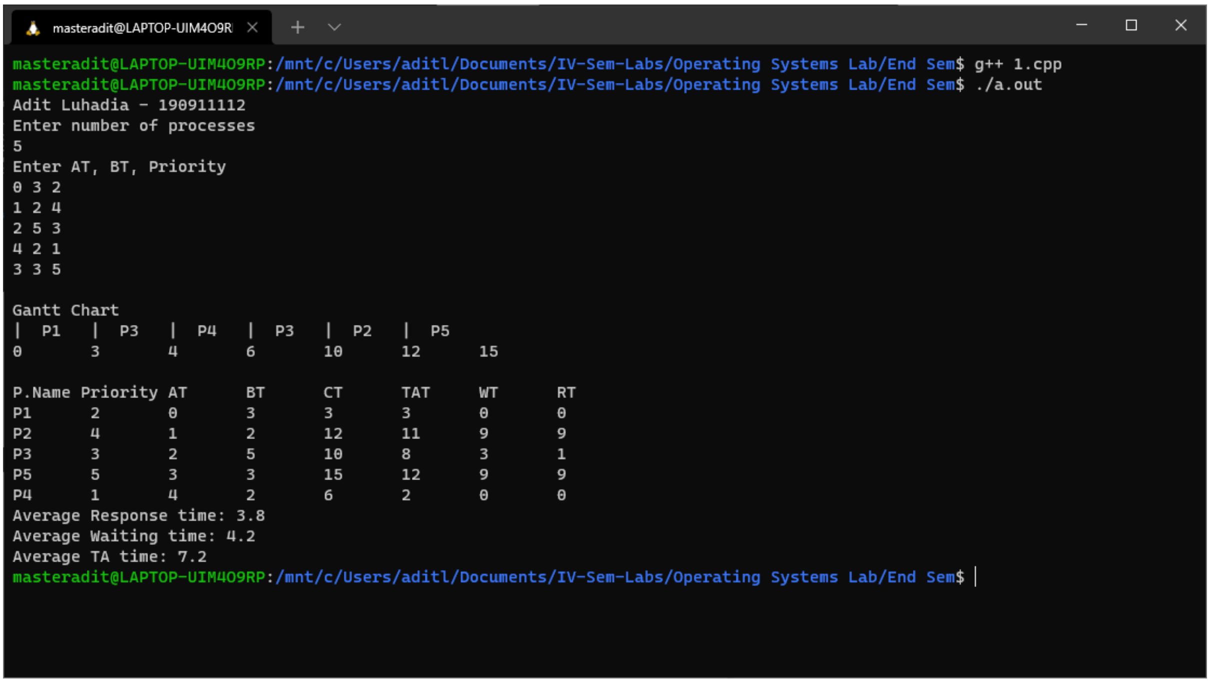
Output screenshots:

Sample run given in the question:

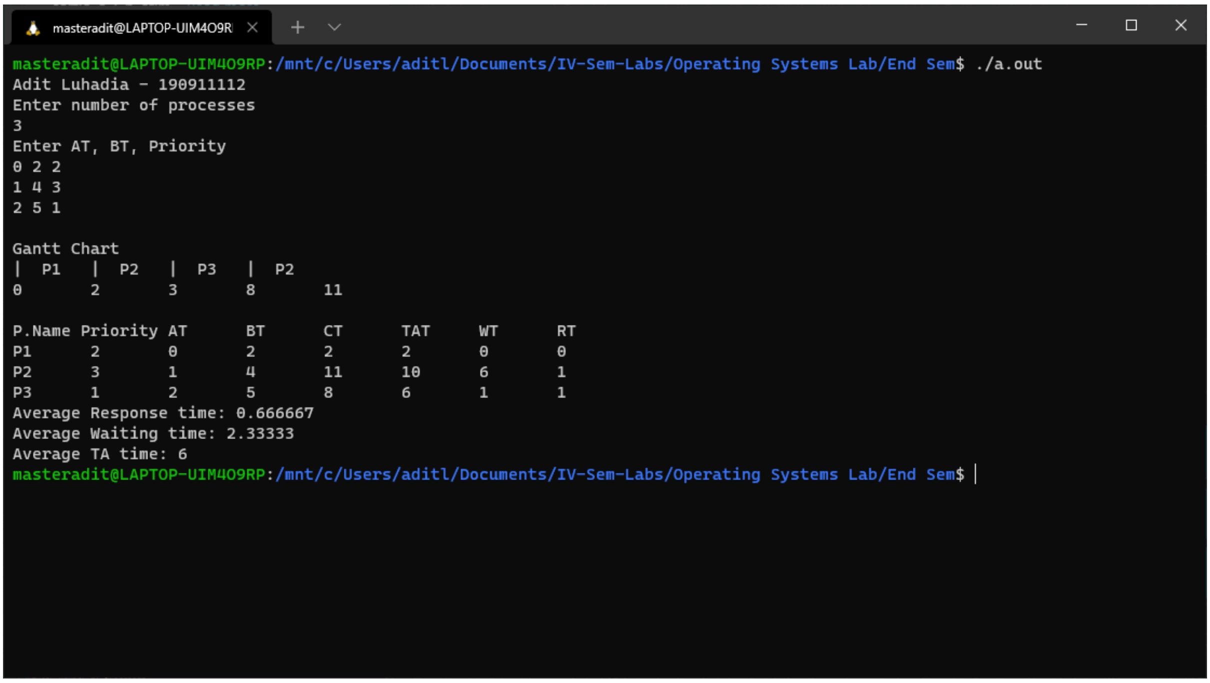


```
masteradit@LAPTOP-UIM4O9RP:/mnt/c/Users/aditl/Documents/IV-Sem-Labs/Operating Systems Lab/End Sem$ g++ 1.cpp
masteradit@LAPTOP-UIM4O9RP:/mnt/c/Users/aditl/Documents/IV-Sem-Labs/Operating Systems Lab/End Sem$ ./a.out
Adit Luhadia - 190911112
Enter number of processes
4
Enter AT, BT, Priority
0 4 4
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-UIM4O9RP:/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-UIM4O9RP:/mnt/c/Users/aditl/Documents/IV-Sem-Labs/Operating Systems Lab/End Sem$ g++ 1.cpp
masteradit@LAPTOP-UIM4O9RP:/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-UIM4O9RP:/mnt/c/Users/aditl/Documents/IV-Sem-Labs/Operating Systems Lab/End Sem$
```



```
masteradit@LAPTOP-UIM4O9RP:/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-UIM4O9RP:/mnt/c/Users/aditl/Documents/IV-Sem-Labs/Operating Systems Lab/End Sem$ |
```