Experiment 10

To implement Job scheduling algorithms.

Name:	**Rishabh Anand**	**UID:**	19BCS4525
Branch:	CSE-IoT	**Sec/Grp:**	1-A
Sem:	6th	**Date:**	12/05/2022
Subject:	CCIoT Lab	**Code:**	CSD-387

1. Aim/Overview of the practical:

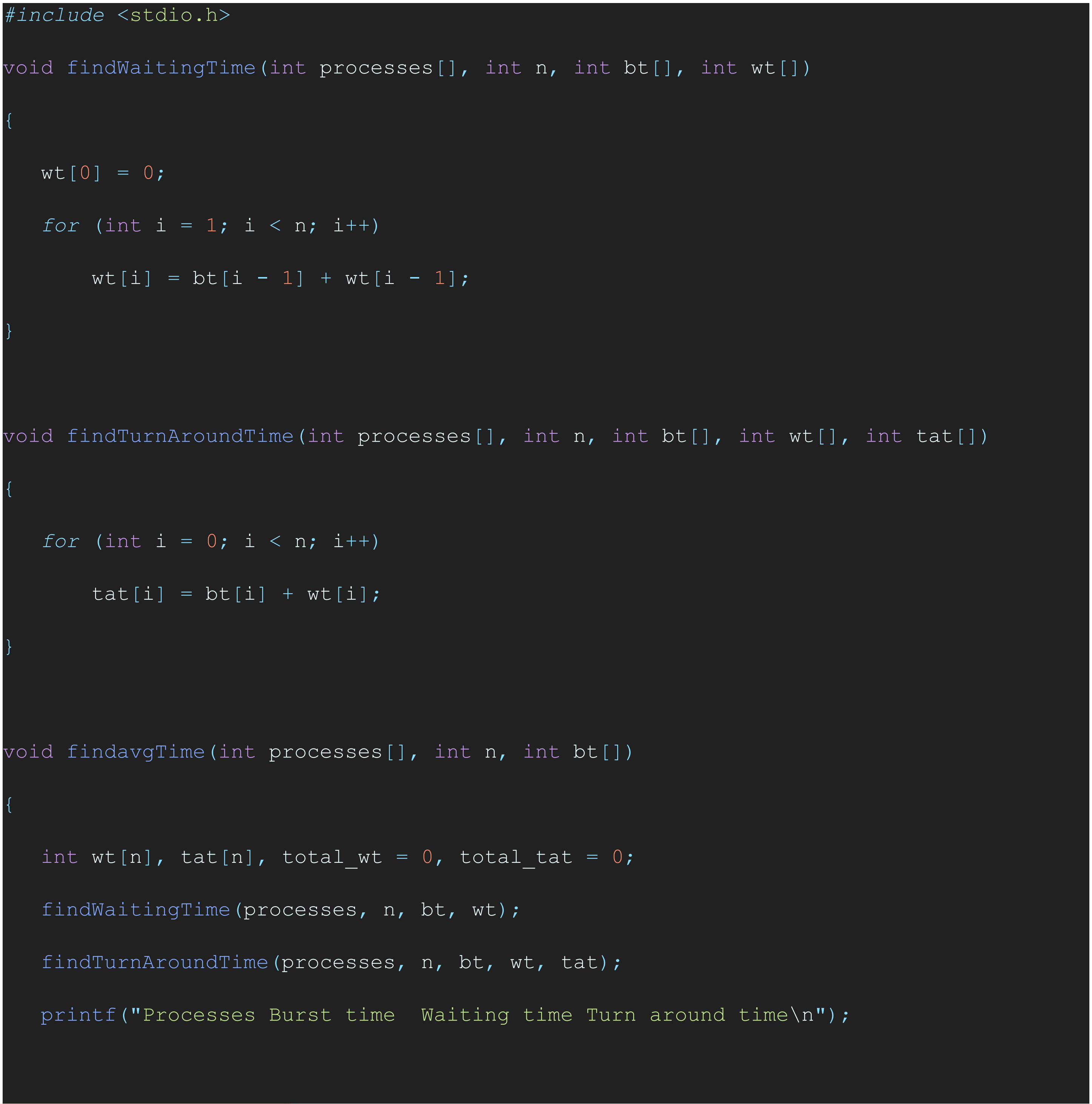
To implement Job scheduling algorithms

2. Task to be done:

- Implementation of FCFS algorithm
- Implementation of min min algorithm

3. Steps for experiment/practical:

CODE FOR FCFS IMPLEMENTATION:

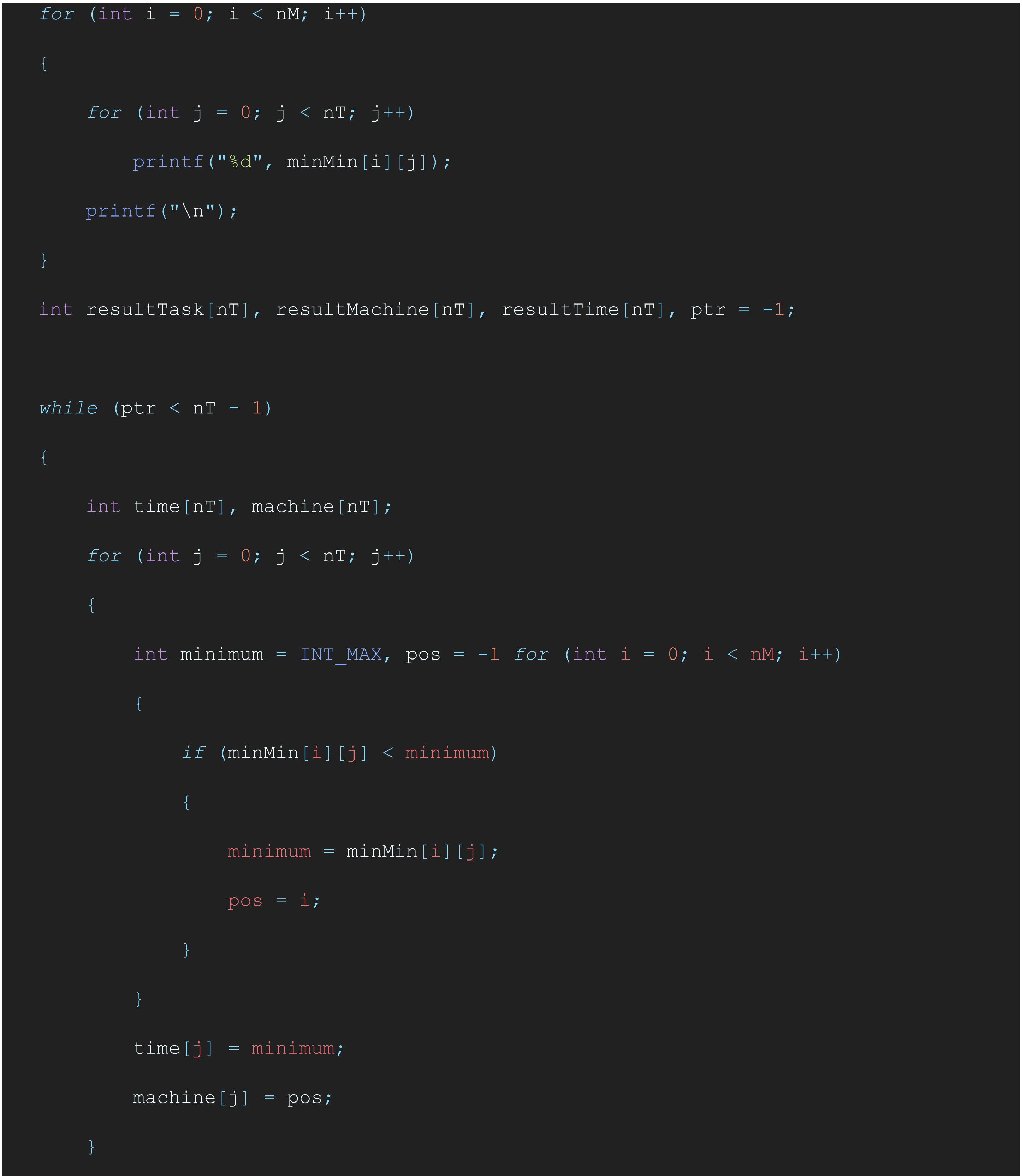


```
#include <stdio.h>

void findWaitingTime(int processes[], int n, int bt[], int wt[])

{

   wt[0] = 0;

   for (int i = 1; i < n; i++)

       wt[i] = bt[i - 1] + wt[i - 1];

}


void findTurnAroundTime(int processes[], int n, int bt[], int wt[], int tat[])

{

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

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

}


void findavgTime(int processes[], int n, int bt[])

{

   int wt[n], tat[n], total_wt = 0, total_tat = 0;

   findWaitingTime(processes, n, bt, wt);

   findTurnAroundTime(processes, n, bt, wt, tat);

   printf("Processes Burst time  Waiting time Turn around time\n");
```

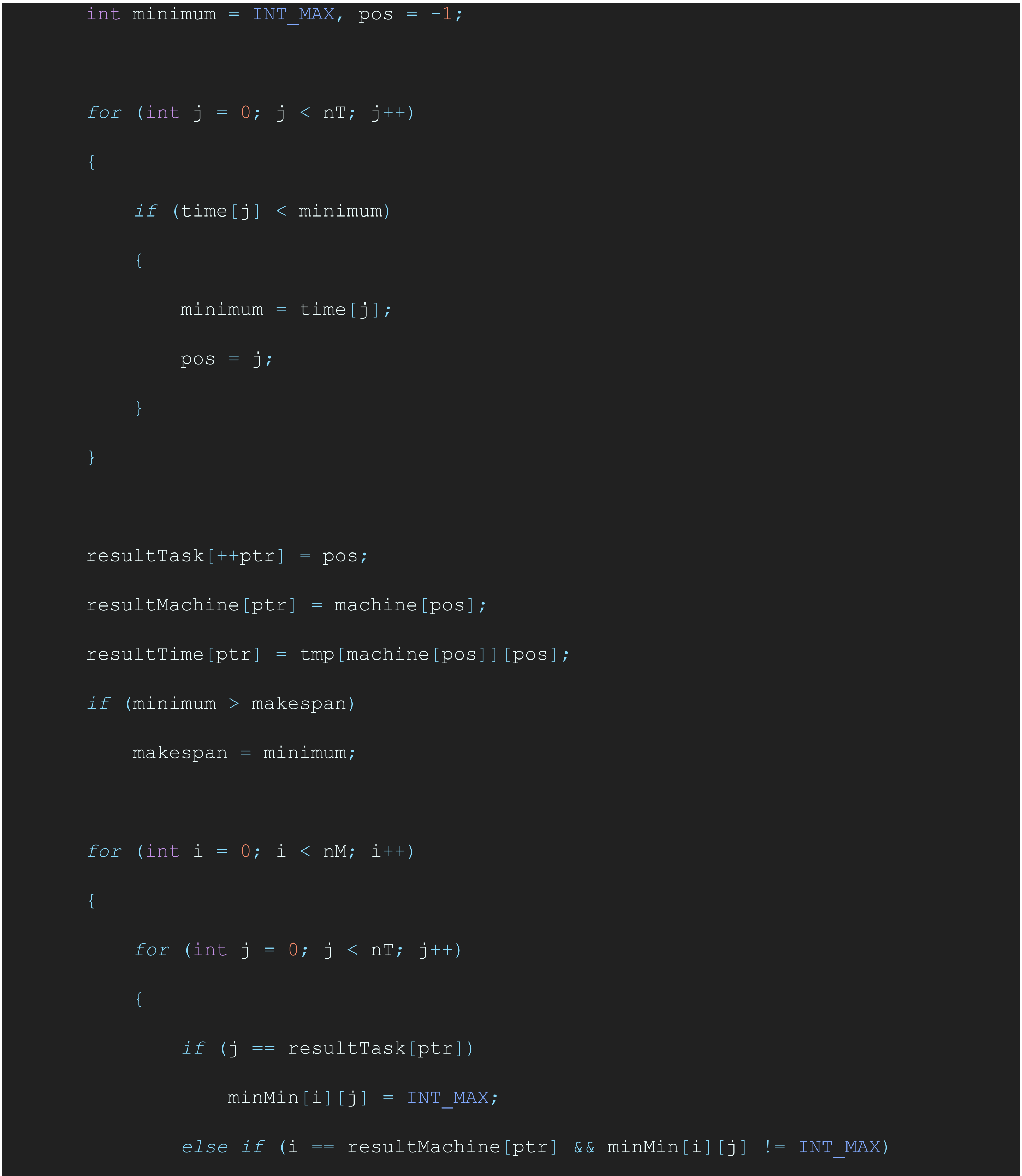
```
    for (int i = 0; i < n; i++)
    {
        total_wt = total_wt + wt[i];
        total_tat = total_tat + tat[i];
        printf(" %d ", (i + 1));
        printf("     %d ", bt[i]);
        printf("     %d", wt[i]);
        printf("     %d\n", tat[i]);
    }

    int s = (float)total_wt / (float)n;
    int t = (float)total_tat / (float)n;
    printf("Average waiting time = %d", s);
    printf("\n");
    printf("Average turn around time = %d ", t);
}
int main()
{
    int processes[] = {1, 2, 3};
    int n = 3;
    int burst_time[] = {10, 5, 8};
    findavgTime(processes, n, burst_time);
    return 0;
}
```

CODE FOR MIN MIN IMPLEMENTATION:

```
#include <stdio.h>
#include <limits.h>

int main()
{
    int nT, nM;
    printf("\nEnter number of machines and tasks\n");
    scanf("%d%d", &nM, &nT);
    int minMin[nM][nT];
    int tmp[nM][nT];
    int makespan = 0;
    printf("\nFill Data\n");
    for (int i = 0; i < nM; i++)
        for (int j = 0; j < nT; j++)
        {
            scanf("%d", &minMin[i][j]);
            tmp[i][j] = minMin[i][j];
        }
    printf("\nOriginal Data \n");
```



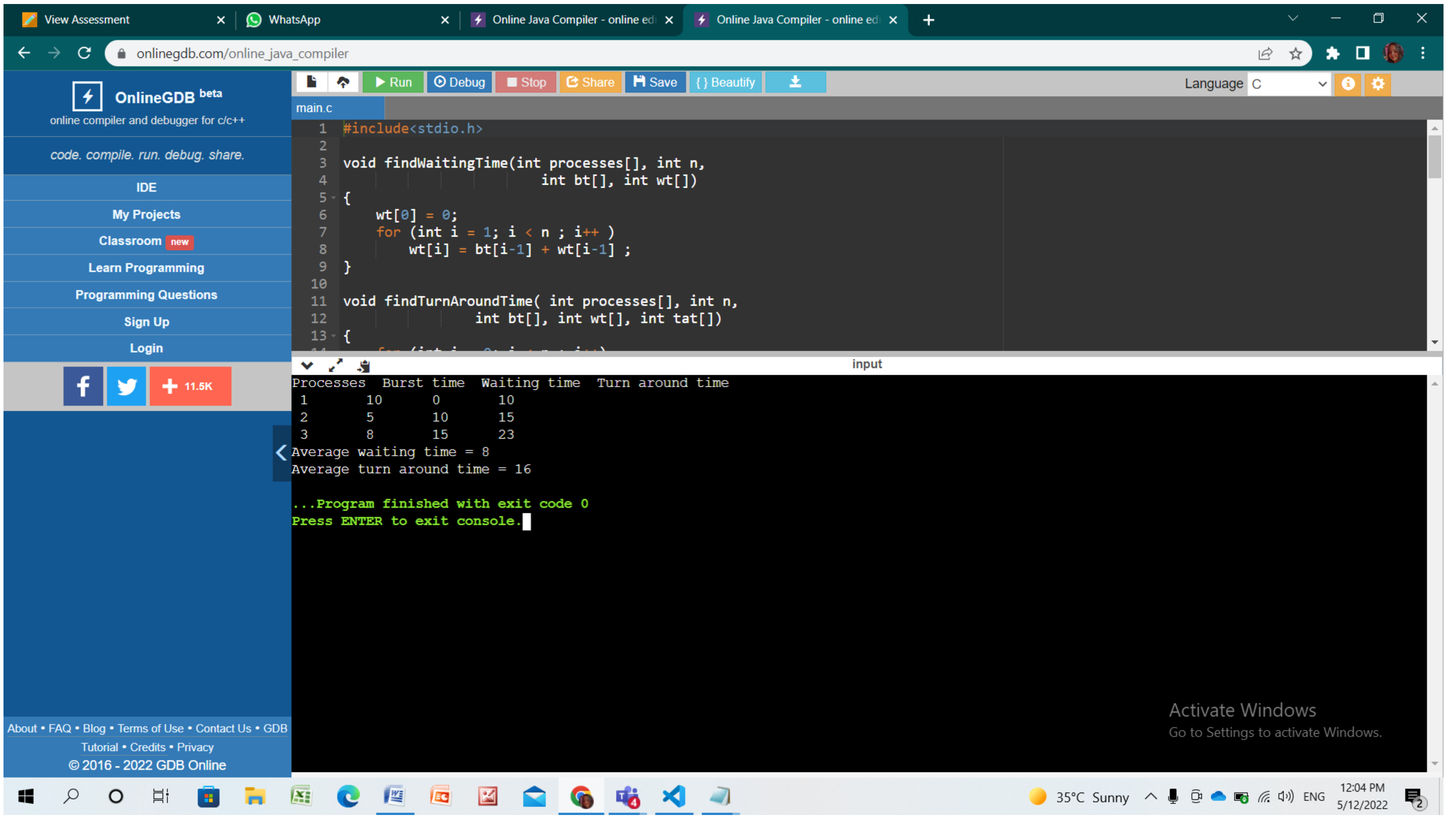
```
for (int i = 0; i < nM; i++)

{

    for (int j = 0; j < nT; j++)

        printf("%d", minMin[i][j]);

    printf("\n");

}

int resultTask[nT], resultMachine[nT], resultTime[nT], ptr = -1;


while (ptr < nT - 1)

{

    int time[nT], machine[nT];

    for (int j = 0; j < nT; j++)

    {

        int minimum = INT_MAX, pos = -1 for (int i = 0; i < nM; i++)

        {

            if (minMin[i][j] < minimum)

            {

                minimum = minMin[i][j];

                pos = i;

            }

        }

        time[j] = minimum;

        machine[j] = pos;

    }
```



```
        int minimum = INT_MAX, pos = -1;


        for (int j = 0; j < nT; j++)

        {

            if (time[j] < minimum)

            {

                minimum = time[j];

                pos = j;

            }

        }


        resultTask[++ptr] = pos;

        resultMachine[ptr] = machine[pos];

        resultTime[ptr] = tmp[machine[pos]][pos];

        if (minimum > makespan)

            makespan = minimum;


        for (int i = 0; i < nM; i++)

        {

            for (int j = 0; j < nT; j++)

            {

                if (j == resultTask[ptr])

                    minMin[i][j] = INT_MAX;

                else if (i == resultMachine[ptr] && minMin[i][j] != INT_MAX)
```

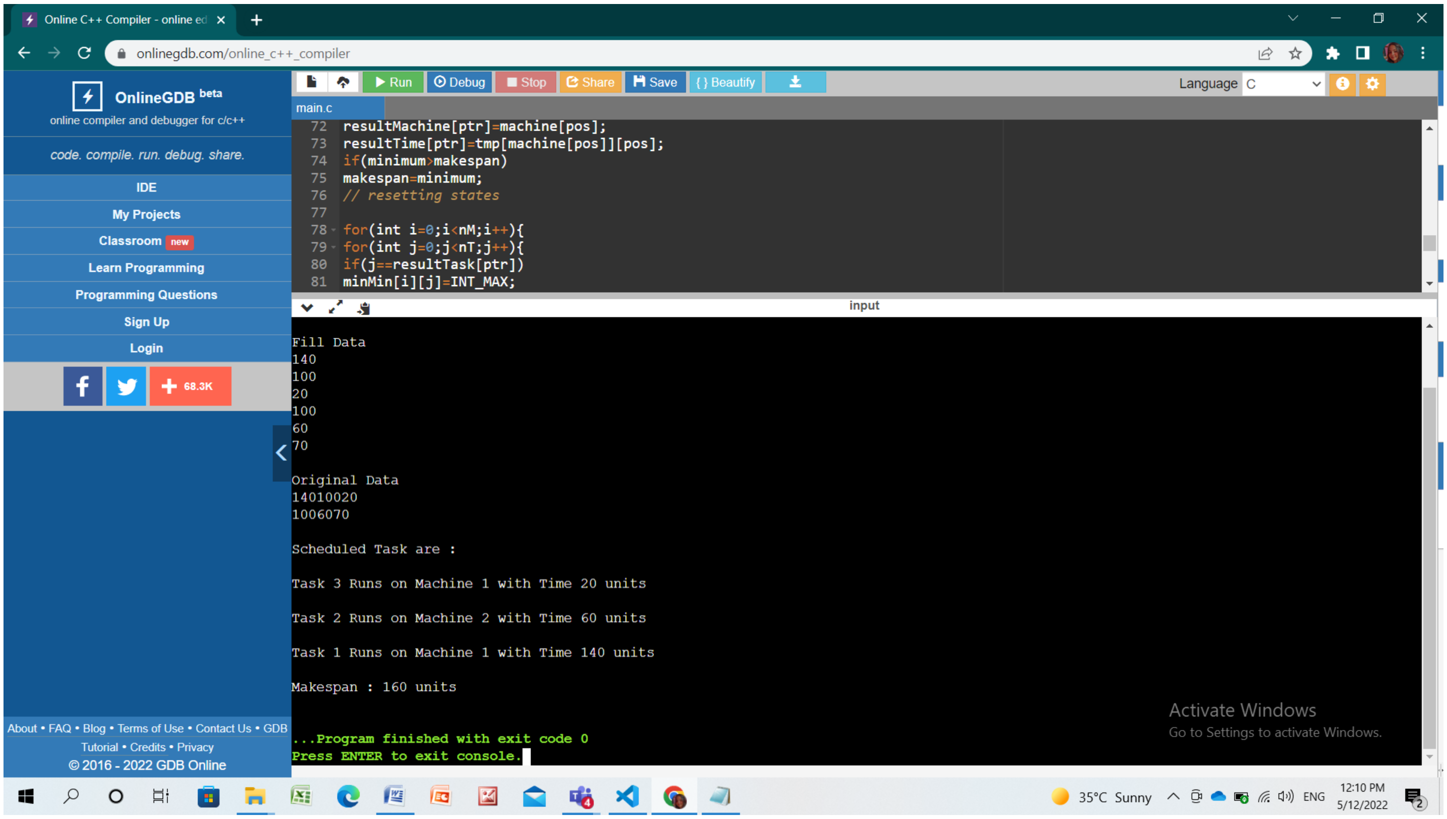
```
                    minMin[i][j] += minimum;
                else
                    continue;
            }
        }
    }
    printf("\nScheduled Task are :\n");
    for (int i = 0; i < nT; i++)
    {
        printf("\nTask %d Runs on Machine %d with Time %d units\n", resultTask[i] + 1,
resultMachine[i] + 1, resultTime[i]);
    }

    printf("\nMakespan : %d units\n", makespan);
    return 0;
}
```

5. Result/Output/Writing Summary:


View Assessment
WhatsApp
Online Java Compiler - online ed
Online Java Compiler - online ed
onlinegdb.com/online_java_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
11.5K
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language C
main.c
input
Activate Windows
Go to Settings to activate Windows.
35°C Sunny
ENG
12:04 PM
5/12/2022

```
#include<stdio.h>

void findWaitingTime(int processes[], int n,
                        int bt[], int wt[])
{
    wt[0] = 0;
    for (int i = 1; i < n ; i++ )
        wt[i] = bt[i-1] + wt[i-1] ;
}

void findTurnAroundTime( int processes[], int n,
                int bt[], int wt[], int tat[])
{
```

```
Processes  Burst time  Waiting time  Turn around time
 1        10       0        10
 2        5        10       15
 3        8        15       23
Average waiting time = 8
Average turn around time = 16

...Program finished with exit code 0
Press ENTER to exit console.
```


Online C++ Compiler - online ed
onlinegdb.com/online_c++_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
68.3K
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language C
main.c
input
Activate Windows
Go to Settings to activate Windows.
35°C Sunny
ENG
12:10 PM
5/12/2022

```
resultMachine[ptr]=machine[pos];
resultTime[ptr]=tmp[machine[pos]][pos];
if(minimum>makespan)
makespan=minimum;
// resetting states

for(int i=0;i<nM;i++){
for(int j=0;j<nT;j++){
if(j==resultTask[ptr])
minMin[i][j]=INT_MAX;
```

```
Fill Data
140
100
20
100
60
70

Original Data
14010020
1006070

Scheduled Task are :

Task 3 Runs on Machine 1 with Time 20 units

Task 2 Runs on Machine 2 with Time 60 units

Task 1 Runs on Machine 1 with Time 140 units

Makespan : 160 units


...Program finished with exit code 0
Press ENTER to exit console.
```

Learning outcomes (What I have learnt):

1. Implementing FCFS ALGORITM
2. Implementing Min Min ALGORITM