Lab Assignment 2
Chaudhary Hamdan
1905387
Date: 21-01-2022

Questions:

1. Min Min Scheduling Algorithm:

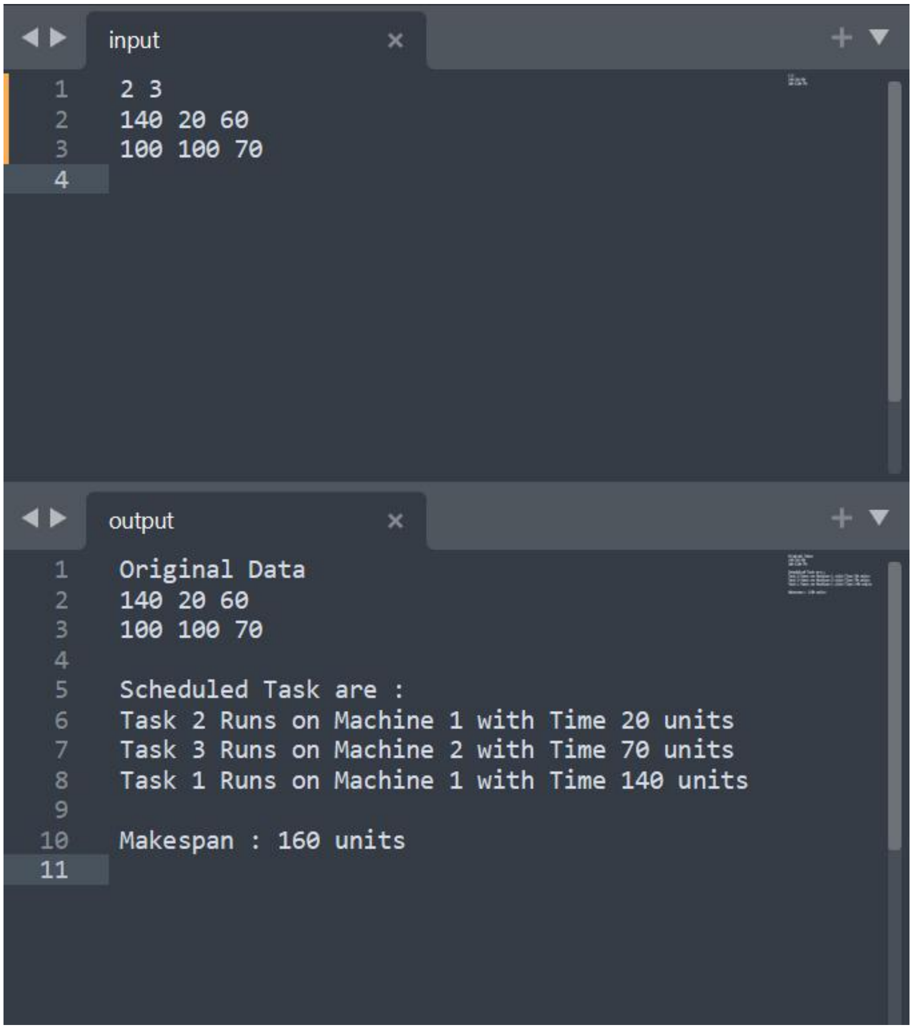
Code:

```
// Author: Chaudhary Hamdan, 1905387
// Generated at: Fri Jan 21 12:47:09 2022

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

int main() {

#ifndef ONLINE_JUDGE
	freopen("C:\\Users\\KIIT\\input", "r", stdin);
	freopen("C:\\Users\\KIIT\\output", "w", stdout);
#endif


	int tasks, machines;
	scanf("%d%d", &machines, &tasks);

	int minMin[machines][tasks];
	int table[machines][tasks];
	int makespan = 0;

	for (int i = 0; i < machines; i++)
		for (int j = 0; j < tasks; j++) {
			scanf("%d", &minMin[i][j]);
			table[i][j] = minMin[i][j];
```

```
    }

printf("Original Data\n");
for (int i = 0; i < machines; i++) {
    for (int j = 0; j < tasks; j++)
        printf("%d ", minMin[i][j]);
    printf("\n");
}

int resultTask[tasks];
int resultMachine[tasks];
int resultTime[tasks];

int ptr = -1;
while (ptr < tasks - 1) {
    int time[tasks], machine[tasks];
    for (int j = 0; j < tasks; j++) {
        int minimum = INT_MAX;
        int pos = -1;
        for (int i = 0; i < machines; i++) {
            if (minMin[i][j] < minimum) {
                minimum = minMin[i][j];
                pos = i;
            }
        }
        time[j] = minimum;
        machine[j] = pos;
    }
    int minimum = INT_MAX;
    int pos = -1;
    for (int j = 0; j < tasks; j++) {
        if (time[j] < minimum) {
            minimum = time[j];
            pos = j;
        }
    }
    resultTask[++ptr] = pos;
    resultMachine[ptr] = machine[pos];
    resultTime[ptr] = table[machine[pos]][pos];
```

```
        if (minimum > makespan)
            makespan = minimum;
        for (int i = 0; i < machines; i++) {
            for (int j = 0; j < tasks; 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 < tasks; i++) {
        printf("Task %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;
}
```

Output:



```
2 3
140 20 60
100 100 70
```

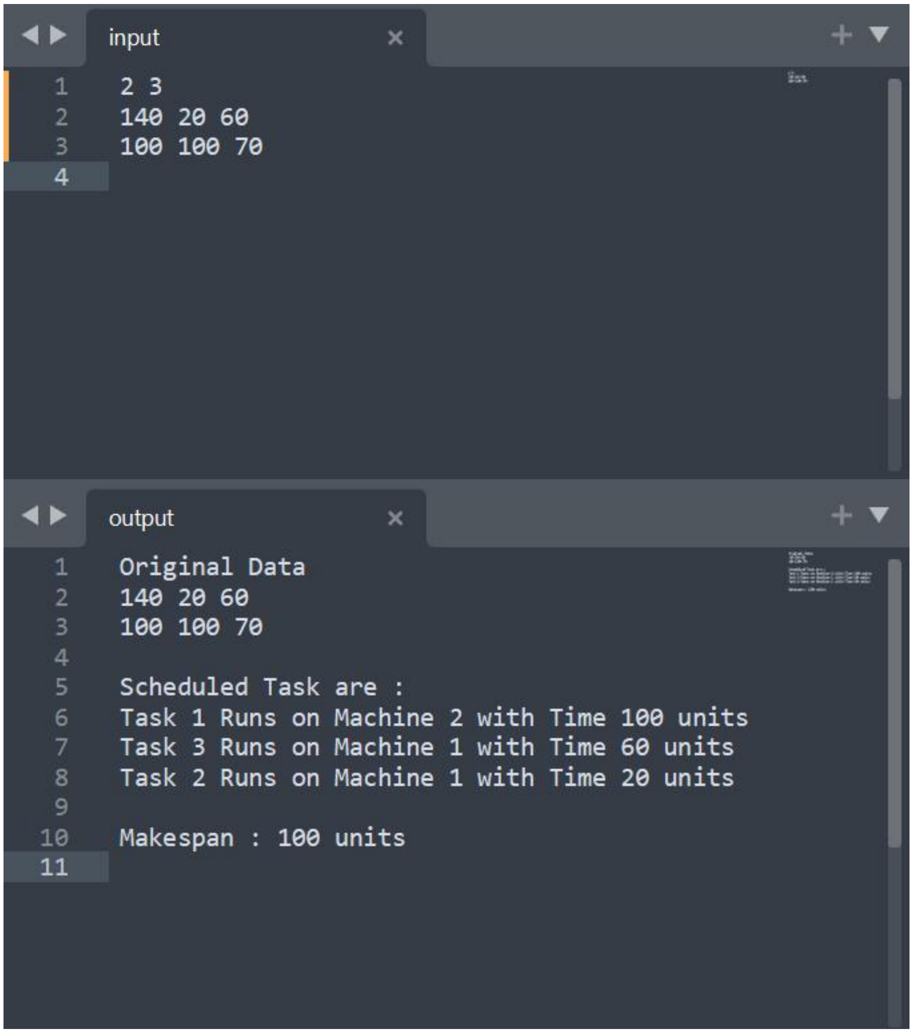
```
Original Data
140 20 60
100 100 70

Scheduled Task are :
Task 2 Runs on Machine 1 with Time 20 units
Task 3 Runs on Machine 2 with Time 70 units
Task 1 Runs on Machine 1 with Time 140 units

Makespan : 160 units
```

2. Max Min Scheduling Algorithm

Code:

```
// Author: Chaudhary Hamdan, 1905387
// Generated at: Fri Jan 21 12:47:25 2022

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

int main() {

#ifndef ONLINE_JUDGE
    freopen("C:\\Users\\KIIT\\input", "r", stdin);
    freopen("C:\\Users\\KIIT\\output", "w", stdout);
#endif


    int tasks, machines;
    scanf("%d%d", &machines, &tasks);

    int maxMin[machines][tasks];
    int table[machines][tasks];
    int makespan = 0;

    for (int i = 0; i < machines; i++)
        for (int j = 0; j < tasks; j++) {
            scanf("%d", &maxMin[i][j]);
            table[i][j] = maxMin[i][j];
        }

    printf("Original Data\n");
    for (int i = 0; i < machines; i++) {
        for (int j = 0; j < tasks; j++)
            printf("%d ", maxMin[i][j]);
        printf("\n");
    }

    int resultTask[tasks];
    int resultMachine[tasks];
    int resultTime[tasks];

    int ptr = -1;
    while (ptr < tasks - 1) {
        int time[tasks], machine[tasks];
        for (int j = 0; j < tasks; j++) {
            int minimum = INT_MAX;
            int pos = -1;
            for (int i = 0; i < machines; i++) {
                if (maxMin[i][j] < minimum) {
                    minimum = maxMin[i][j];
                    pos = i;
                }
            }
            time[j] = minimum;
            machine[j] = pos;
        }
        int maximum = INT_MIN;
        int pos = -1;
        for (int j = 0; j < tasks; j++) {
            if (time[j] > maximum && time[j] != INT_MAX) {
                maximum = time[j];
                pos = j;
            }
        }
        resultTask[++ptr] = pos;
        resultMachine[ptr] = machine[pos];
        resultTime[ptr] = table[machine[pos]][pos];
        if (maximum > makespan)
            makespan = maximum;
        for (int i = 0; i < machines; i++) {
            for (int j = 0; j < tasks; j++) {
                if (j == resultTask[ptr])
                    maxMin[i][j] = INT_MAX;

                else if (i == resultMachine[ptr] && maxMin[i][j] !=
INT_MAX)
                    maxMin[i][j] += maximum;
                else
                    continue;
                }
            }
        }
        printf("\nScheduled Task are :\n");
        for (int i = 0; i < tasks; i++) {
            printf("Task %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;
}
```

Output:



```
2 3
140 20 60
100 100 70
```

```
Original Data
140 20 60
100 100 70

Scheduled Task are :
Task 1 Runs on Machine 2 with Time 100 units
Task 3 Runs on Machine 1 with Time 60 units
Task 2 Runs on Machine 1 with Time 20 units

Makespan : 100 units
```