Group Project:

CLO: 4	**SO: I**

Programming Checkpoint 2: **Due Date: Sunday October 13, 2019 till 11:59 PM**

Make the following changes and additions in your Checkpoint 1 solution.

1. Instead of reading input from keyboard, write code so that entire input is read from a file named "**input.txt**".
2. Instead of displaying output on screen, display the output in the file "**output.txt**".

If the **input** file contains the following commands:

WEIGHTED (This string shows that the graph is weighted)

10 (This integer indicates number of vertices. So, your program should create 10 vertices, that is 0 for 'a', 1 for 'b' and so on 9 for 'j'.

12 (This integer indicates number of edges. The program must read next 12 lines for source, target and weight of each edge)

0 2 10 (First integer indicates source vertex, second integer indicates target vertex and third integer indicates the weight of this edge)

0 3 5

0 4 2

1 4 3

1 5 2

2 3 1

2 5 3

4 5 15

6 7 5

6 9 4

7 8 10

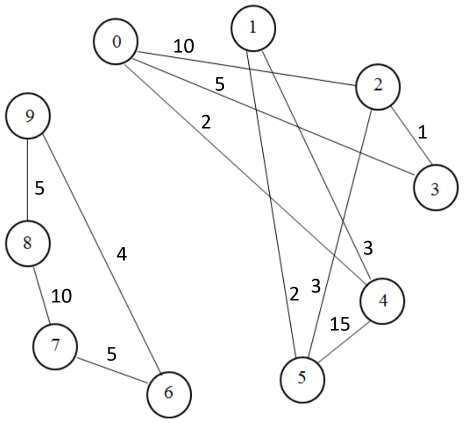
8 9 5

QUIT

The output of the program should be in **output file** output.txt as follows:

Weight Matrix:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	0	0	10	5	2	0	0	0	0	0
1	0	0	0	0	3	2	0	0	0	0
2	10	0	0	1	0	3	0	0	0	0
3	5	0	1	0	0	0	0	0	0	0
4	2	3	0	0	0	15	0	0	0	0
5	0	2	3	0	15	0	0	0	0	0
6	0	0	0	0	0	0	0	5	0	4
7	0	0	0	0	0	0	5	0	10	0
8	0	0	0	0	0	0	0	10	0	5
9	0	0	0	0	0	0	4	0	5	0



Adjacent Vertices of every vertex:

VERTEX: 0 {a} - VISIT: false - ADJACENCY: 3,2,4
VERTEX: 1 {b} - VISIT: false - ADJACENCY: 5,4
VERTEX: 2 {c} - VISIT: false - ADJACENCY: 0,3,5
VERTEX: 3 {d} - VISIT: false - ADJACENCY: 0,2
VERTEX: 4 {e} - VISIT: false - ADJACENCY: 5,1,0
VERTEX: 5 {f} - VISIT: false - ADJACENCY: 2,4,1
VERTEX: 6 {g} - VISIT: false - ADJACENCY: 7,9
VERTEX: 7 {h} - VISIT: false - ADJACENCY: 6,8
VERTEX: 8 {i} - VISIT: false - ADJACENCY: 7,9
VERTEX: 9 {j} - VISIT: false - ADJACENCY: 8,6

DFS traversal: 0, 2, 3, 5, 1, 4, 6, 7, 8, 9

BFS traversal: 0, 2, 3, 4, 5, 1, 6, 7, 9, 8

Graph is not connected.

If the **input** file contains the following commands:

UNWEIGHTED (This shows that the graph is unweighted)

10 (This integer indicates number of vertices. So, your program should create 10 vertices, that is 0 for ‘a’, 1 for ‘b’ and son on 9 for ‘j’.

12 (This integer indicates number of edges. The program must read next 11 lines for source, target and weight of each edge)

0 2 (First integer indicates source vertex, and second integer indicates target vertex. The program should store 0 in the weight data member for all edges)

0 3

0 4

1 4

1 5

2 3

2 5

4 5

6 7

6 9

7 8

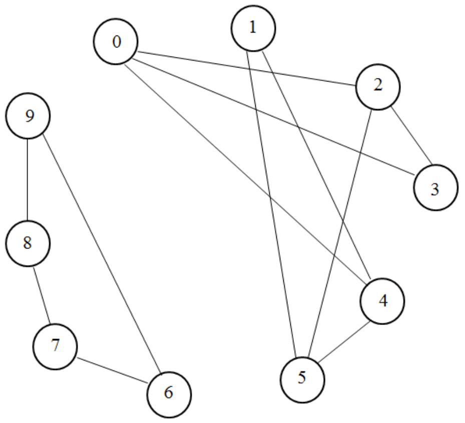
8 9

QUIT

The output of the program should be in **output file** as follows:

Adjacency Matrix:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	0	0	1	1	1	0	0	0	0	0
1	0	0	0	0	1	1	0	0	0	0
2	1	0	0	1	0	1	0	0	0	0
3	1	0	1	0	0	0	0	0	0	0
4	1	1	0	0	0	1	0	0	0	0
5	0	1	1	0	1	0	0	0	0	0
6	0	0	0	0	0	0	0	1	0	1
7	0	0	0	0	0	0	1	0	1	0
8	0	0	0	0	0	0	0	1	0	1
9	0	0	0	0	0	0	1	0	1	0



Adjacent Vertices of every vertex:

VERTEX: 0 {a} - VISIT: false - ADJACENCY: 3,2,4
VERTEX: 1 {b} - VISIT: false - ADJACENCY: 5,4

```
VERTEX: 2 {c} - VISIT: false - ADJACENCY: 0,3,5
VERTEX: 3 {d} - VISIT: false - ADJACENCY: 0,2
VERTEX: 4 {e} - VISIT: false - ADJACENCY: 5,1,0
VERTEX: 5 {f} - VISIT: false - ADJACENCY: 2,4,1
VERTEX: 6 {g} - VISIT: false - ADJACENCY: 7,9
VERTEX: 7 {h} - VISIT: false - ADJACENCY: 6,8
VERTEX: 8 {i} - VISIT: false - ADJACENCY: 7,9
VERTEX: 9 {j} - VISIT: false - ADJACENCY: 8,6
```

DFS traversal: 0, 2, 3, 5, 1, 4, 6, 7, 8, 9

BFS traversal: 0, 2, 3, 4, 5, 1, 6, 7, 9, 8

Graph is not connected.

Distribution of Tasks:

For the checkpoint 2, the distribution of tasks for each and every group member is given below:

Member 1: will write the code for reading the input from input.txt file.
Member 2: will write the code for writing the output into output.txt file.
Member 3: will write the code to display the output in the given format.

Note:

Only **Member 1** of every group will **upload** the solution on Blackboard always.

Group project (Checkpoint 2) has been uploaded in “Group Project” option. It is a group project. Its due date is **Sunday October 13, 2019 till 11:59 PM**. You can also upload the solution by **Monday October 14, 2019 till 11:59 PM with 25% deduction of marks and by Tuesday October 15, 2019 till 11:59 PM with 50% deduction of marks**. **After this date, no submission will be accepted.**