III. You have been given two sorted integer arrays of size m and n. Design an algorithm and implement it using a program to find list of elements which are common to both. (Time Complexity = O(m+n))

Input Format:
First line contains m (the size of first array).
Second line contains m space-separated integers describing first array.
Third line contains n (the size of second array).
Fourth line contains n space-separated integers describing second array.

Output Format:
Output will be the list of elements which are common to both.

Sample I/O Problem III:

Input:	**Output:**

7
34 76 10 39 85 10 55
12
30 55 34 72 10 34 10 89 11 30 69 51
	10 10 34 55

Note: Consider the following input format in the form of adjacency matrix for graph based questions (directed/undirected/weighted/unweighted graph).

Input Format: Consider example of below given graph in Figure (a).
A boolean matrix AdjM of size V X V is defined to represent edges of the graph. Each edge of graph is represented by two vertices (start vertex u, end vertex v). That means, an edge from u to v is represented by making AdjM[u,v] and AdjM[v,u] = 1. If there is no edge between u and v then it is represented by making AdjM[u,v] = 0. Adjacency matrix representation of below given graph is shown in Figure (b). Hence edges are taken in the form of adjacency matrix from input. In case of weighted graph, an edge from u to v having weight w is represented by making AdjM[u,v] and AdjM[v,u] = w.

Input format for this graph is shown in Figure (c).
First input line will obtain number of vertices V present in graph.
After first line, V input lines are obtained. For each line i in V, it contains V space separated boolean integers representing whether an edge is present between i and all V.

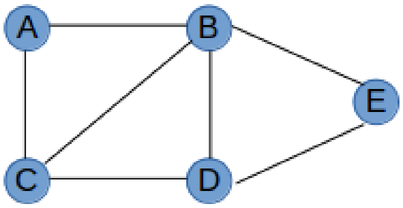
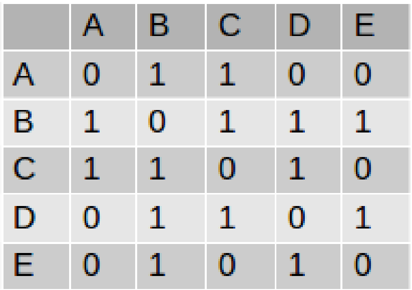
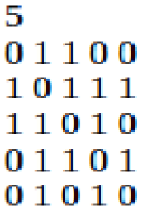


Figure (a)



	A	B	C	D	E
A	0	1	1	0	0
B	1	0	1	1	1
C	1	1	0	1	0
D	0	1	1	0	1
E	0	1	0	1	0

Figure (b)



```
5
0 1 1 0 0
1 0 1 1 1
1 1 0 1 0
0 1 1 0 1
0 1 0 1 0
```

Figure (c)

Week 6:

I. Given a (directed/undirected) graph, design an algorithm and implement it using a program to find if a path exists between two given vertices or not. (Hint: use DFS)

Input Format:
Input will be the graph in the form of adjacency matrix or adjacency list.
Source vertex number and destination vertex number is also provided as an input.

Output Format:
Output will be '**Yes Path Exists**' if path exists, otherwise print '**No Such Path Exists**'.

Sample I/O Problem I:

Input:	**Output:**

5
0 1 1 0 0
1 0 1 1 1
1 1 0 1 0
0 1 1 0 1
0 1 0 1 0
1 5
	Yes Path Exists

II. Given a graph, design an algorithm and implement it using a program to find if a graph is bipartite or not. (Hint: use BFS)

Input Format:
Input will be the graph in the form of adjacency matrix or adjacency list.

Output Format:
Output will be '**Yes Bipartite**' if graph is bipartite, otherwise print '**Not Bipartite**'.

Sample I/O Problem II:

Input:	**Output:**

5
0 1 1 0 0
1 0 1 1 1
1 1 0 1 0
0 1 1 0 1
0 1 0 1 0
	Not Bipartite

III. Given a directed graph, design an algorithm and implement it using a program to find whether cycle exists in the graph or not.

Input Format:
Input will be the graph in the form of adjacency matrix or adjacency list.

Output Format:
Output will be '**Yes Cycle Exists**' if cycle exists otherwise print '**No Cycle Exists**'.

Sample I/O Problem III:

Input:	**Output:**

5
0 1 1 0 0
0 0 0 1 1
	No Cycle Exists

0 1 0 1 0
0 0 0 0 1
0 0 0 0 0	

Week 7:

Note: Input, output format along with sample input output for problem I and II is same and is provided at the end of problem II.

I. After end term examination, Akshay wants to party with his friends. All his friends are living as paying guest and it has been decided to first gather at Akshay’s house and then move towards party location. The problem is that no one knows the exact address of his house in the city. Akshay as a computer science wizard knows how to apply his theory subjects in his real life and came up with an amazing idea to help his friends. He draws a graph by looking in to location of his house and his friends’ location (as a node in the graph) on a map. He wishes to find out shortest distance and path covering that distance from each of his friend’s location to his house and then whatsapp them this path so that they can reach his house in minimum time. Akshay has developed the program that implements Dijkstra’s algorithm but not sure about correctness of results. Can you also implement the same algorithm and verify the correctness of Akshay’s results? (Hint: Print shortest path and distance from friends’ location to Akshay’s house)

II. Design an algorithm and implement it using a program to solve previous question's problem using Bellman- Ford's shortest path algorithm.

Input Format:
Input will be the graph in the form of adjacency matrix or adjacency list.
Source vertex number is also provided as an input.

Output Format:
Output will contain V lines.
Each line will represent the whole path from destination vertex number to source vertex number along with minimum path weigth.

Sample I/O Problem I and II:

Input: 5 0 4 1 0 0 0 0 0 0 4 0 2 0 4 0 0 0 0 0 4 0 0 0 0 0 1	**Output:** 1 : 0 2 3 1 : 3 3 1 : 1 4 3 1 : 3 5 2 3 1 : 7

III. Given a directed graph with two vertices (source and destination). Design an algorithm and implement it using a program to find the weight of the shortest path from source to destination with exactly k edges on the path.

Input Format:
First input line will obtain number of vertices V present in the graph.
Graph in the form of adjacency matrix or adjacency list is taken as an input in next V lines.