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:	**Output:**

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
	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.

Next input line will obtain source and destination vertex number.
Last input line will obtain value k.

Output Format:
Output will be the weigth of shortest path from source to destination having exactly k edges.
If no path is available then print “**no path of length k is available**”.

Sample I/O Problem III:

Input:	**Output:**

4
0 10 3 2
0 0 0 7
0 0 0 6
0 0 0 0
1 4
2
	Weight of shortest path from (1,4) with 2 edges : 9

Week 8:

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. Assume that a project of road construction to connect some cities is given to your friend. Map of these cities and roads which will connect them (after construction) is provided to him in the form of a graph. Certain amount of rupees is associated with construction of each road. Your friend has to calculate the minimum budget required for this project. The budget should be designed in such a way that the cost of connecting the cities should be minimum and number of roads required to connect all the cities should be minimum (if there are N cities then only N-1 roads need to be constructed). He asks you for help. Now, you have to help your friend by designing an algorithm which will find minimum cost required to connect these cities. (use Prim's algorithm)

II. Implement the previous problem using Kruskal's algorithm.

Input Format:
The first line of input takes number of vertices in the graph.
Input will be the graph in the form of adjacency matrix or adjacency list.

Output Format:
Output will be minimum spanning weight

Sample I/O Problem I and II:

Input:	**Output:**

7
0 0 7 5 0 0 0
0 0 8 5 0 0 0
7 8 0 9 7 0 0
5 0 9 0 15 6 0
0 5 7 15 0 8 9
0 0 0 6 8 0 11
0 0 0 0 9 11 0
	Minimum Spanning Weight: 39