Lab Task#11: Find the shortest path in a weighted graph

1. You are given a directed graph, a source from which you will start your journey and a target for which you have to print the path. You have to find the minimum cost/distance required to arrive at each of the reachable vertices and print the path as given in the output section in a sequence.
    a. First will have the total number of nodes (**n**) and the total number of edges (**m**).
        i. Nodes are numbered from **1** to **n**
        ii. Next m lines will be followed by m pairs of integers denoting the directed edges.
        iii. a b w
            1. It means there is a connection from **a to b** and
            2. The cost/distance from **a to b** is **w**
    b. Then two integers **s** and **t** denoting the **source** node and the **target** node.
2. Outputs:
    a. Print the minimum distance required to reach a particular node from the source in a sequence from **1 to n**.
    b. Avoid printing the distance for the source node.
    c. If a node is not reachable. Mention that.
    d. Print the **path for the target** node only.
    e. See the output format for more details
3. Use the idea of **SSSP** to solve the problem.
4. Try to solve the problem in paper first to understand how you might approach it.
5. Implementation should be done in either C or C++ or Python or Java or javascript.
    a. Explain your code in words if possible.
    b. Also, if I ask you about your code, you better be able to answer. So please, understand the code before submitting it.
6. **Assignment File Name**: AlgoLabAssign**11**_SSSP-NNweights_**191**-115-***ZZZ***
    a. Replace ***ZZZ*** with your roll.
7. Related material: https://youtu.be/COM74cDxmp4
8. If You find any problem in the question, let me know. I will correct it.

Input #1	***Output#1***
4 5 1 2 1	Minimum distances from Source 1 to other nodes 2 1

1 3 4 2 3 2 2 4 6 3 4 3 **1 4**	3 3 4 6 Path to the target node: **4** Path taken: 1 2 3 4
Input #2	***Output#2***
5 7 2 4 3 2 5 6 1 5 1 3 4 7 3 2 2 3 1 6 4 5 5 **3 4**	Minimum distances from Source 3 to other nodes 1 6 2 2 4 5 5 7 Path to the target node: **4** Path taken: 3 2 4
Input #3	***Output#3***
5 10 1 2 10 1 4 5 2 3 1 2 4 2 3 5 4 4 2 3 4 3 9 4 5 2 5 1 7 5 3 6 **1 3**	*Minimum distances from Source 1 to other nodes* *2 8* *3 9* *4 5* *5 7* *Path to the target node:* ***3*** *Path taken: 1 4 2 3*
Input #4	***Output#4***
5 9 1 2 4 1 3 2 2 4 2 2 3 3 2 5 3 3 2 1 3 5 5 3 4 4 5 4 1 **1 5**	Minimum distances from Source 1 to other nodes 2 3 3 2 4 5 5 6 Path to the target node: **5** Path taken: 1 3 2 5