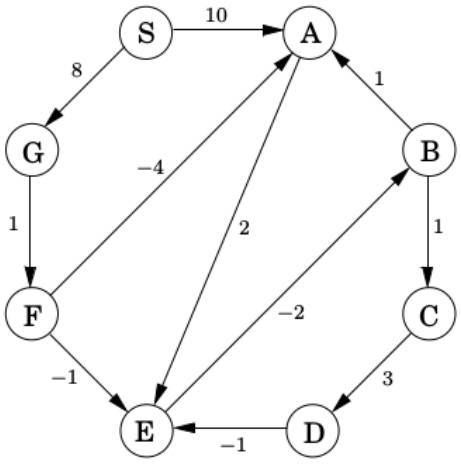
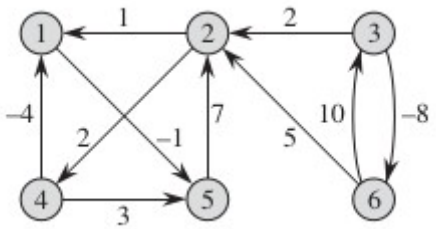
Lab Assignment 11 (28 December 2020)

Problem 1 : Implement the Bellman-Ford algorithm for computing the shortest paths from a source vertex s to all other vertices in a graph. Consider your input graph to be a weighted directed graph where edge weights might be negative. Print the distance table for each iteration, as shown below. If the input graph has negative weight cycles, your code should detect this and print a message saying 'Graph has a negative weight cycle' . For e.g. in the graph shown below (source DPV), if the weight of edge (C,D) is changed to 1, a negative weight cycle is formed.



	Iteration							
Node	0	1	2	3	4	5	6	7
S	0	0	0	0	0	0	0	0
A	∞	10	10	5	5	5	5	5
B	∞	∞	∞	10	6	5	5	5
C	∞	∞	∞	∞	11	7	6	6
D	∞	∞	∞	∞	∞	14	10	9
E	∞	∞	12	8	7	7	7	7
F	∞	∞	9	9	9	9	9	9
G	∞	8	8	8	8	8	8	8

Problem 2: Implement Floyd-Warshall's algorithm to compute the shortest path between all pairs of vertices in a graph. You could test your algorithm on the graph shown below (source CLRS):



Problem 3: Write a program to print all possible valid ways to break a sentence without spaces into a sequence of valid words. Use a dictionary to store the set of valid words. For e.g. if the dictionary is {a, an, at, the, are, man, hunt, go, ant, he, mango}

Sample Run:

Enter the sentence without spaces: anthehuntmango
The possible sequences of words are:
ant he hunt mango
an the hunt mango
ant he hunt man go
an the hunt man go

Problem 4: Implement the dynamic-programming algorithm to solve the Knapsack Problem.