**2) Describe how you would update the random\_graph method to generate a graph G = (V,E) that does not contain a negative weight cycle. You are given a function that can determine whether or not an edge completes a negative weight cycle.**

The random\_graph function will generate a graph randomly using <limits.h> file with Int\_Min and Int\_Max values. The weights interval can be **[0, 2w] instead of [-w, w]**. The given function checks for |V|-1 times and if it can find a shortest path even after |V|-1 times, then there exists a negative weight cycle. A cycle is considered as negative weight cycle, when there is a bigger number with negative sign. It will be keep going and weights will be negative infinity for all vertices on the cycle.

A negative cycle can be there only if it is not reachable from the source.

**Example of weights in a cyclic graph:**

1. Weight1 = 2; Weight2 = 3; Weight3 =-6 => 2+3+(-6) => -1 – Contains a negative cycle.
2. Weight1 = 2; Weight2 = -3; Weight3 =6 => 2+(-3)+6 => 5 – Doesn’t contain a negative cycle.

**3) What is the running time for Bellman-Ford using an adjacency matrix representation?**

The running time for Bellman-Ford algorithm using an adjacency matrix representation is O(VE) as it runs for all the vertices and all the edges. It is a single source shortest path algorithm in which S is the source and all the vertices are visited from S.