**Problem:**

1. Write a program (with explanatory comments) to generate a random graph (simple) of order *n* and density ρ (). Test it with . Submit the program listing, generated graphs (adjacency matrix & sketch) and your observation.

1. Write a program to compute the degree sequence of a given graph (sorted non-increasing order). Submit the output.

**SOLUTION FOR PROBLEM 1 & 2:**

I have developed a program that accepts 2 inputs (the number of vertices and the density of the graph) to print the ADJACENCY MATRIX of the graph.

The same code also prints the DEGREE SEQUENCE of the graph. (Sorted vertex-vice)

{EXAMPLE SEQUENCE -> deg (vert1), deg (vert2), deg (vert3)}

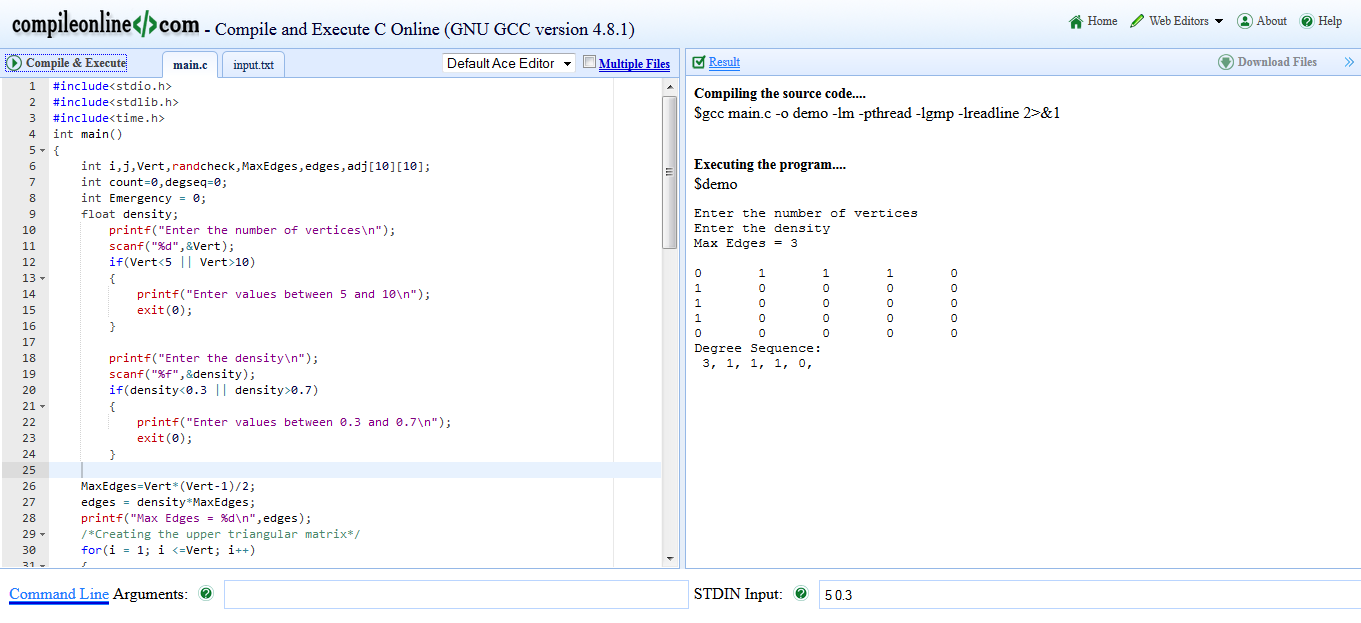
For the execution please read the “Readme.txt” file.

**Screen Shots of a few trials:**

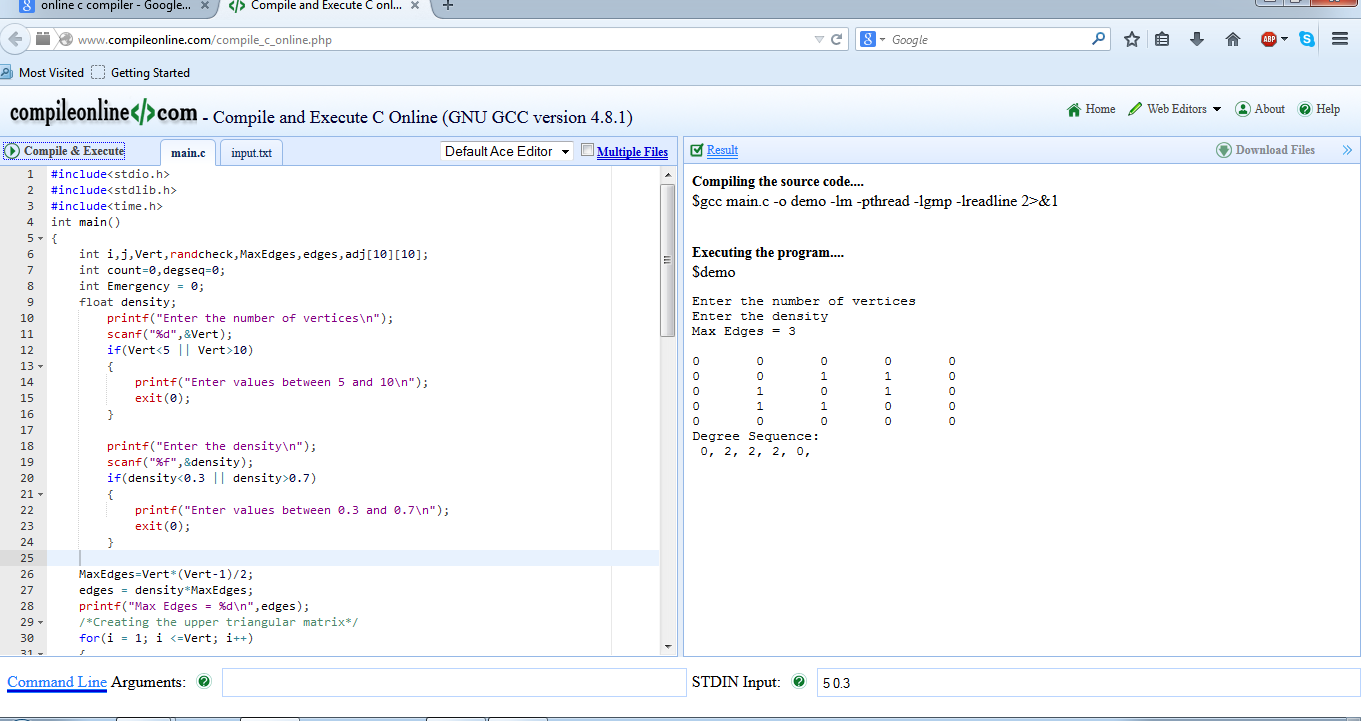
**CASE 1:**

**INPUTS:** NUMBER OF VERTICES = 5 DENSITY = 0.3

**TRIAL 1:**



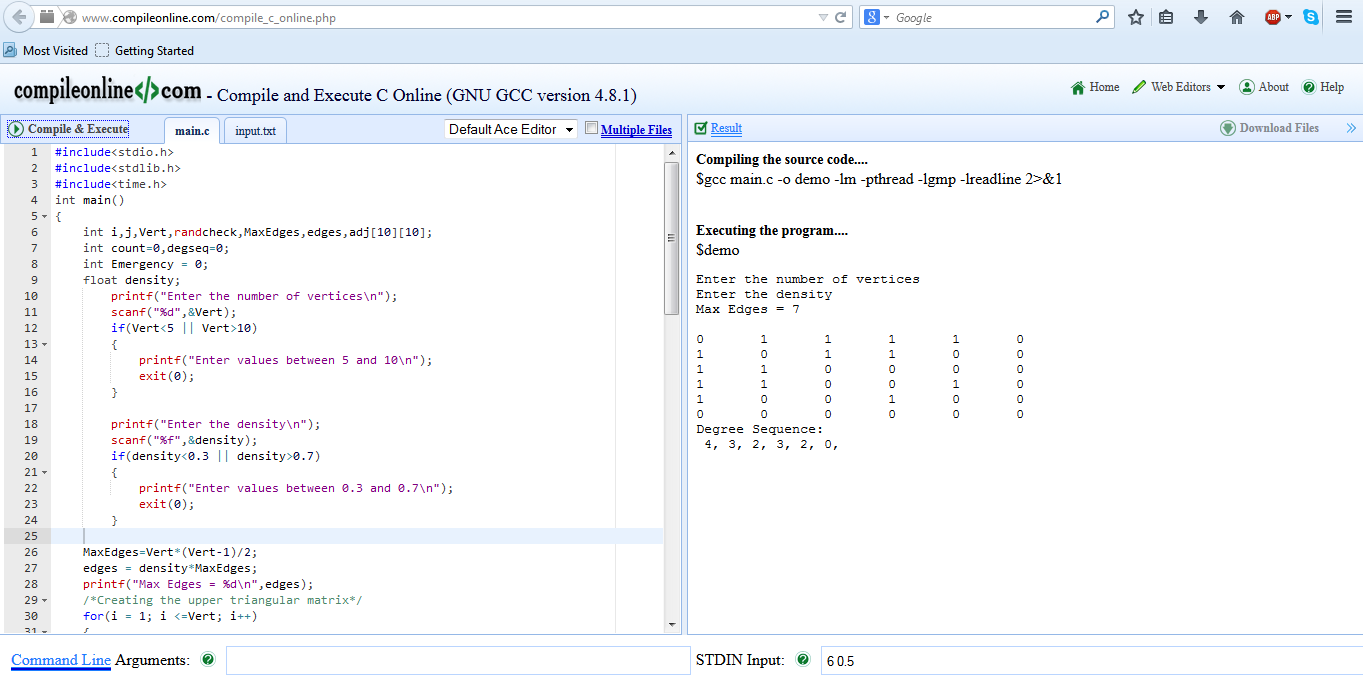
**TRIAL 2**



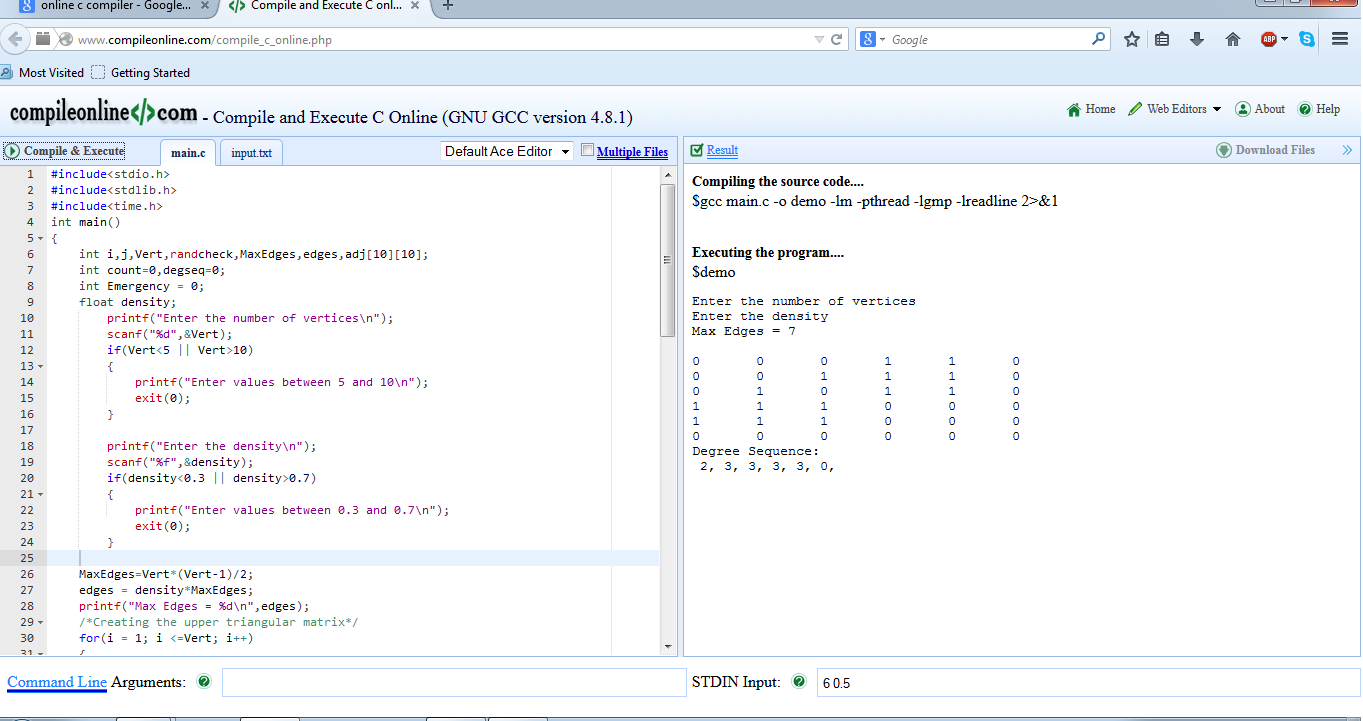
**CASE 2**

**INPUTS:** NUMBER OF VERTICES = 6 DENSITY = 0.5

**TRIAL 1:**



**TRIAL 2:**



**OBSERVATIONS:**

* The number of edges will be an integer value and it is calculated using the two inputs. (FORMULA given in the problem)
* The number of 1’s in the adjacency matrix will be twice the number of edges that has been generated. (Upper triangular and the lower triangular matrix will be Symmetric. If 2->3, it also means that 3->2)
* The edges are generated randomly between random vertices and hence the degree sequence might vary for various inputs.
* The randomness is applied while picking the vertices before connecting them using a random edge.
* The code can be optimized by reducing the loops and using an array of pointers!