Programming assignment 9.
Due date: Tuesday, May 7, 2019 at 11:00pm

In this program you are required to implement a random undirected graph and determine if the graph is bipartite or not using *BFS*. Here, we work with three colors for the vertices: *gray* (not visited), [*blue, red*] (**opposite colors**)

1. Request the user to determine the order (|V|) and size (|E|) of the graph.
2. Generate **|E|** random edges into the adjacency matrix/list (Adj) to make a random undirected graph. (Make sure to have a **symmetric** matrix)
3. Print the resulting adjacency matrix/list.
4. Implement 2 functions: *Explore* and *Is_bipartite*
5. In ***Explore*** function,
   a. For each vertex (v) initialize *v.color = "gray".*
   b. Start from the first vertex and call *Is_bipartite* on that.
   c. Next, go to the next unexplored vertex (having *gray* color), and call *Is_bipartite* again.
   d. Repeat *step c.* until every vertex is explored/colored or a not bipartite graph is detected.
6. Now to implement our second function (*Is_bipartite*), you need to change your BFS function in lab 8.
   a. Keep popping each vertex from **Q**. (call it **u**)
   b. Go to the adjacency list of **u**, (adj(**u**)), and **for** each neighbor (**v**):
   c. If ***v***. *color* == "*gray*", assign an opposite color to **v** and push it into the **Q**. (**Example**: ***u***.*color* is *blue*, and ***v***.*color* is *gray* ➔ we set ***v***.*color* = "*red*")
   d. Else if ***v***.*color* == ***u***.*color*: Stop the entire code and print "**NOT bipartite**".
7. Print the color of all the vertices.