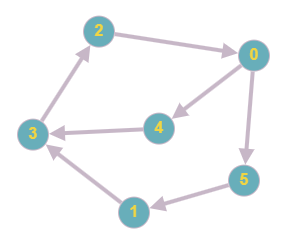
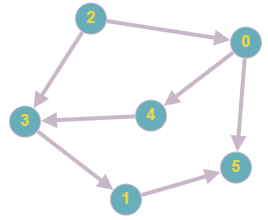
**Question 1 [15 Points]**

You need to write a function called **reverseEdge().** That takes a **directed graph** represented as an adjacency matrix in its parameter. Your task is to reverse all the outgoing connections from the ***odd*** vertices in the graph and return the matrix.

| **Sample Given Adjacency Matrix** | | **Sample Output Adjacency Matrix** | |
| --- | --- | --- | --- |
|  | **0, 0, 0, 0, 1, 1,**  **0, 0, 0, 1, 0, 0,**  **1, 0, 0, 0, 0, 0,**  **0, 0, 1, 0, 0, 0,**  **0, 0, 0, 1, 0, 0,**  **0, 1, 0, 0, 0, 0,** |  | **0, 0, 0, 0, 1, 1,**  **0, 0, 0, 0, 0, 1,**  **1, 0, 0, 1, 0, 0,**  **0, 1, 0, 0, 0, 0,**  **0, 0, 0, 1, 0, 0,**  **0, 0, 0, 0, 0, 0,** |
| **Sample Function Call** | | | |
| **reverseEdge( graph )** | | | |

**Explanation:** Each row in the adjacency matrix represents a Vertex, and the columns represent other Vertices where the outgoing connection is made. For example, in the sample input, the 1th row (0 0 0 1 0 0) means there is an outgoing edge from Vertex 1 to Vertex 3. After reversing the connection the outgoing edge would be going from Vertex 3 to Vertex 1. This is seen in the Row 3 ( 0 1 0 0 0 0 ) of the sample output. Moreover, the 3th row (0 0 1 0 0 0) means there is an outgoing edge from Vertex 3 to Vertex 2. After reversing the connection the outgoing edge would be going from Vertex 2 to Vertex 3. This is seen in the Row 2 ( 1 0 0 1 0 0 ) of the sample output. This reversing is only applicable for the ***odd*** vertices like 1,3,5.