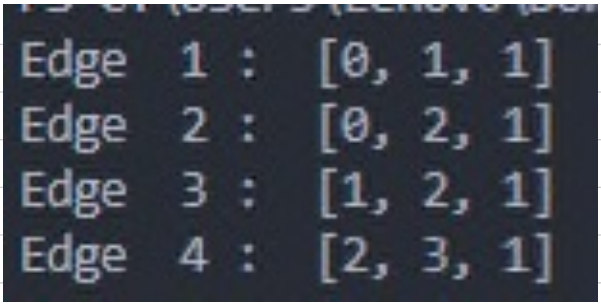

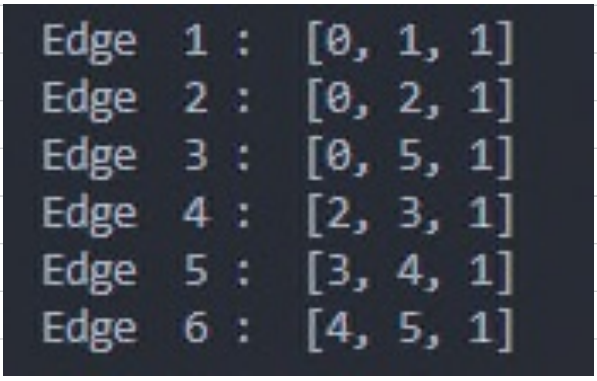

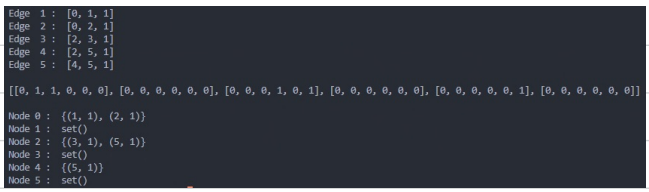

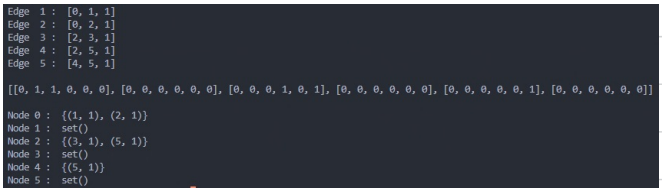

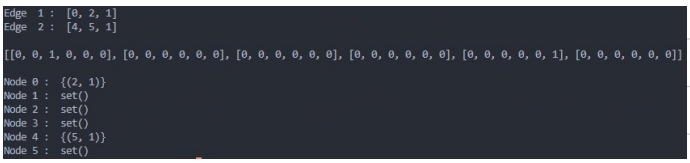


```
class Graph:
    def __init__(self, nodeNum, directed=True):
        self.myNodeIP = nodeNum
        self.rangeNodes = range(self.myNodeIP)

        self.edgesList = []
        self.myMatrix = [[0 for column in
range(nodeNum)]
                         for row in range(nodeNum)]

        self.myDirected = directed
        self.adjList = {node: set() for node in
self.rangeNodes}

    def addEdge(self, node1, node2, connector):
        self.edgesList.append([node1, node2,
connector])
        self.myMatrix[node1][node2] = connector
        self.adjList[node1].add((node2,
connector))

        if not self.myDirected:
            self.edgesList.append([node1, node2,
connector])
            self.myMatrix[node2][node1] =
connector
            self.adjList[node2].add((node1,
connector))

    def edgeList(self):
        edgeCount = len(self.edgesList)
        for i in range(edgeCount):
            print("Edge ", i+1, ": ", self.edgesList[i])
        print()

    def matrix(self):
        print(self.myMatrix)
        print()

    def myList(self):
        for key in self.adjList.keys():
            print("Node", key, ": ", self.adjList[key])


graph = Graph(6)


graph.addEdge(0, 2, 1)
graph.addEdge(0, 4, 1)
graph.addEdge(1, 2, 1)
graph.addEdge(3, 5, 1)
graph.addEdge(4, 5, 1)


graph.edgeList()
graph.matrix()
graph.myList()
```