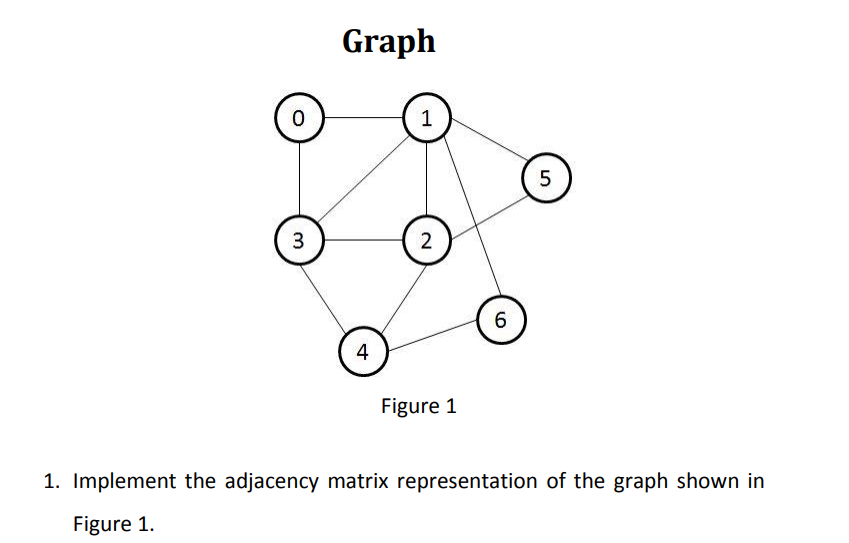
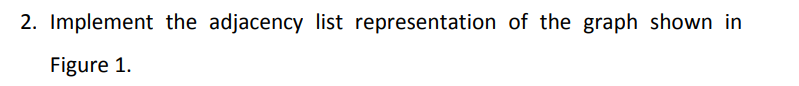
**S Abhishek AM.EN.U4CSE19147**

**Data Structures**

****

class Graph(object):  
  
 def \_\_init\_\_(self**,** row\_col):  
 self.arr = []  
 for i in range(row\_col):  
 self.arr.append([**0** for i in range(row\_col)])  
 self.row\_col = row\_col  
  
 def add(self**,** v1**,** v2):  
 if v1 == v2:  
 self.arr[v1][v2] = **1** self.arr[v2][v1] = **1** def display(self):  
 for row in self.arr:  
 for val in row:  
 print('{}'.format(val))  
 print()  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 graph = Graph(**7**)  
 graph.add(**0, 1**)  
 graph.add(**0, 3**)  
 graph.add(**1, 2**)  
 graph.add(**2, 3**)  
 graph.add(**3, 4**)  
 graph.add(**4, 6**)  
  
 graph.display()



class Node:  
 def \_\_init\_\_(self**,** value):  
 self.vertex = value  
 self.next = None  
  
  
class Graph:  
 def \_\_init\_\_(self**,** data):  
 self.V = data  
 self.graph = [None] \* self.V  
  
 def add(self**,** x**,** y):  
 node = Node(y)  
 node.next = self.graph[x]  
 self.graph[x] = node  
  
 node = Node(x)  
 node.next = self.graph[y]  
 self.graph[y] = node  
  
 def display(self):  
 for i in range(self.V):  
 ptr = self.graph[i]  
 while ptr:  
 print(" -> {}".format(ptr.vertex))  
 ptr = ptr.next  
 print()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 graph = Graph(**7**)  
 graph.add(**0, 1**)  
 graph.add(**0, 3**)  
 graph.add(**1, 2**)  
 graph.add(**2, 3**)  
 graph.add(**2, 4**)  
 graph.add(**3, 4**)  
 graph.add(**4, 6**)  
  
 graph.display()