DAY-7:

GRAPHS:

PROGRAM:

class GraphMatrix:

def \_\_init\_\_(self, num\_vertices):

self.num\_vertices = num\_vertices

self.matrix = [[0 for \_ in range(num\_vertices)] for \_ in range(num\_vertices)]

def add\_edge(self, u, v):

if 0 <= u < self.num\_vertices and 0 <= v < self.num\_vertices:

self.matrix[u][v] = 1

self.matrix[v][u] = 1

else:

print("Invalid vertex!")

def display(self):

print("Adjacency Matrix:")

for row in self.matrix:

print(row)

g = GraphMatrix(7)

g.add\_edge(0, 1)

g.add\_edge(0, 2)

g.add\_edge(1, 3)

g.add\_edge(2, 3)

g.add\_edge(3, 4)

g.add\_edge(4, 5)

g.add\_edge(5, 6)

g.display()

OUTPUT:  
Adjacency Matrix:

[0, 1, 1, 0, 0, 0, 0]

[1, 0, 0, 1, 0, 0, 0]

[1, 0, 0, 1, 0, 0, 0]

[0, 1, 1, 0, 1, 0, 0]

[0, 0, 0, 1, 0, 1, 0]

[0, 0, 0, 0, 1, 0, 1]

[0, 0, 0, 0, 0, 1, 0]

MATRIX:

PROGRAM:

class GraphAdjMatrix:

def \_\_init\_\_(self,size=None,adj\_matrix=None):

if adj\_matrix is not None:

self.adj\_matrix=adj\_matrix

self.size=len(adj\_matrix)

else:

self.size=size

self.adj\_matrix=[[0 for \_ in range(size)] for \_ in range(size)]

def add\_edge(self,u,v):

if u < self.size and v < self.size:

self.adj\_matrix[u][v]=1

self.adj\_matrix[v][u]=1

def display(self):

print("Adjacency Matrix")

for row in self.adj\_matrix:

print(row)

adj\_matrix=[

[0,1,1,0,0,0,0],

[1,0,0,1,0,0,0],

[1,0,0,1,0,0,0],

[0,1,1,0,1,0,0],

[0,0,0,1,0,1,0],

[0,0,0,0,1,0,1],

[0,0,0,0,0,1,0]

]

g = GraphAdjMatrix(adj\_matrix=adj\_matrix)

g.display()

OUTPUT:

[0, 1, 1, 0, 0, 0, 0]

[1, 0, 0, 1, 0, 0, 0]

[1, 0, 0, 1, 0, 0, 0]

[0, 1, 1, 0, 1, 0, 0]

[0, 0, 0, 1, 0, 1, 0]

[0, 0, 0, 0, 1, 0, 1]

[0, 0, 0, 0, 0, 1, 0]