Q3.

1. **BFS**

  def BFS(self, vertex, graph):

        visited = [False] \* (max(self.graph) + 1)

        queue = []

         visited[vertex] = True

        while queue:

             s = queue.pop(0)

            print (s, end = " ")

            for i in self.graph[vertex]:

                if visited[i] == False:

                    queue.append(i)

                    visited[i] = True

if(visited[vertex] = true):

if(curr == vertex):

print(“Eulerian cycle exist”)

1. **DFS**

def DFS(self, v, graph):

        visited = set()

        visited.add(v)

        # Recur for all the vertices

        # adjacent to this vertex

        for neighbour in self.graph[v]:

            if neighbour not in visited:

                self.DFS(neighbour, visited)

if(final node == v ):

print(“Eulerian cycle Exist”)

1. **Eulerian Cycle:**

