AIM:

The aim is to solve the deapth first search program using python program

PROGRAM:

from collections import defaultdict

class Graph:

def \_\_init\_\_(self):

self.graph = defaultdict(list)

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

self.graph[u].append(v)

def DFSUtil(self, v, visited):

visited.add(v)

print(v, end=' ')

for neighbor in self.graph[v]:

if neighbor not in visited:

self.DFSUtil(neighbor, visited)

def DFS(self, v):

visited = set()

self.DFSUtil(v, visited)

g = Graph()

g.add\_edge(0, 1)

g.add\_edge(0, 2)

g.add\_edge(1, 2)

g.add\_edge(2, 3)

g.add\_edge(3, 4)

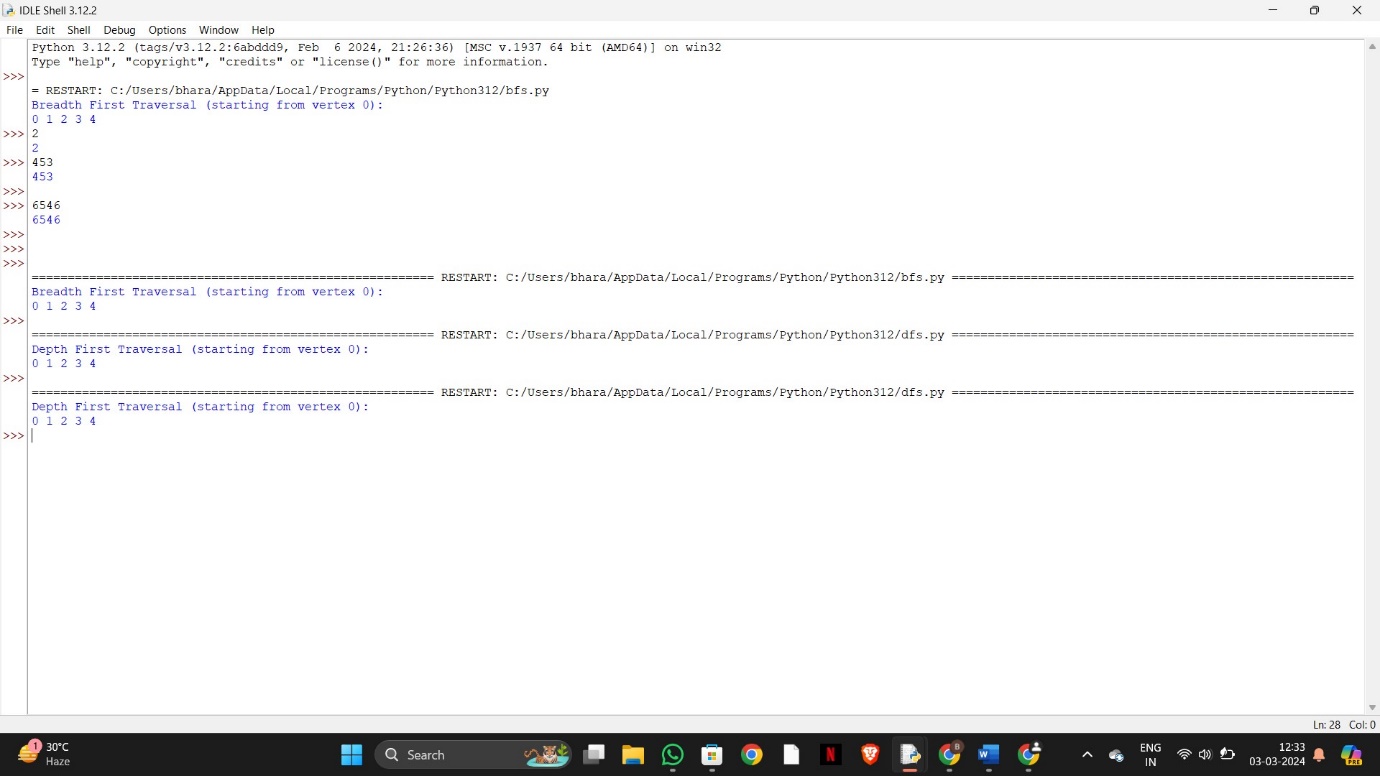
print("Depth First Traversal (starting from vertex 0):")

g.DFS(0)

INPUT:

Deapth first traversal (starting from vertex 0):

OUTPUT:



RESULT:

To solving deapth first search using python programming is successfully completed