AIM:

The aim is to solve the breadth first search 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 BFSUtil(self, v, visited):

visited.add(v)

queue = [v]

while queue:

v = queue.pop(0)

print(v, end=' ')

for neighbor in self.graph[v]:

if neighbor not in visited:

queue.append(neighbor)

visited.add(neighbor)

def BFS(self, v):

visited = set()

self.BFSUtil(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)

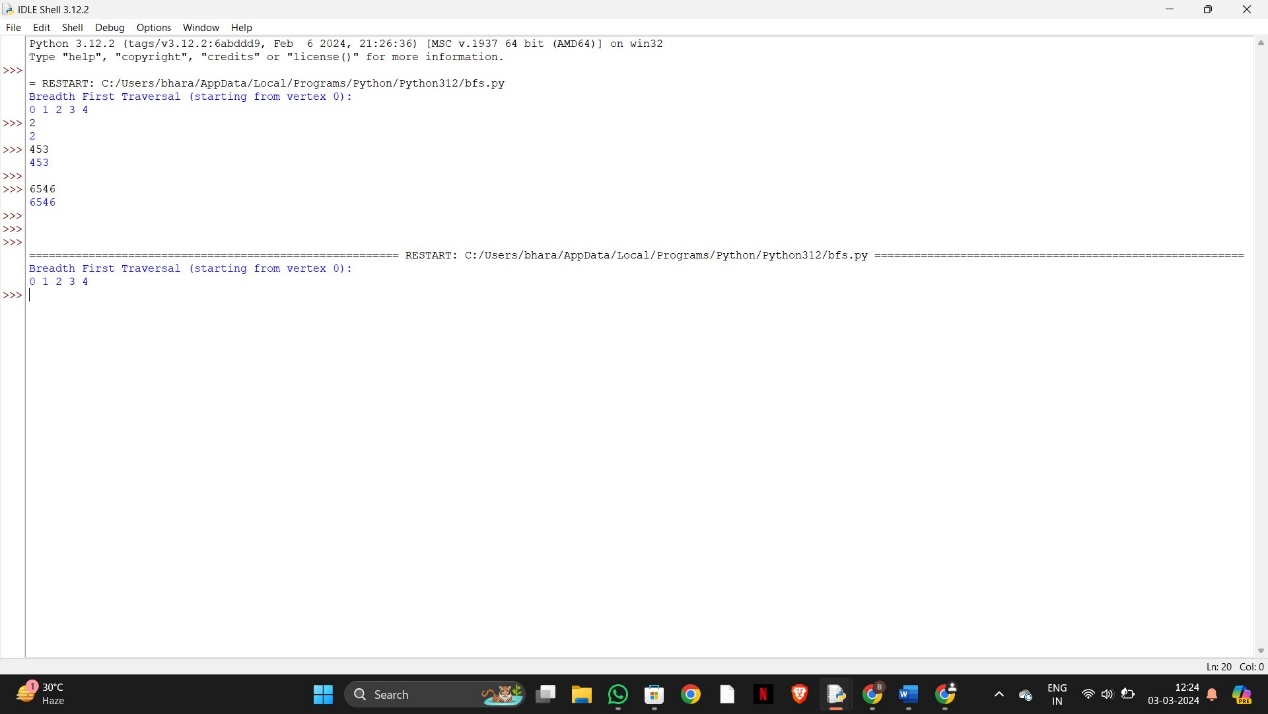
print("Breadth First Traversal (starting from vertex 0):")

g.BFS(0)

INPUT:

Breadth first search traversal (starting from vertex 0):0,1,2,3,4

OUTPUT:



RESULT:

To solve the breadth first search using python program is successfully completed