**SOFTWARE LABORATORY 1**

**GROUP B – EXPERIMENT 1**

**TITLE:**

Implement depth first search algorithm and Breadth First Search algorithm. Use an undirected graph and develop a recursive algorithm for searching all the vertices of a graph or tree data structure.

**CODE:**

from collections import deque

# New graph as adjacency list

graph = {

'1': ['2', '3'],

'2': ['1', '4'],

'3': ['1', '5'],

'4': ['2'],

'5': ['3', '6', '7'],

'6': ['5'],

'7': ['5']

}

# DFS - Recursive using call stack

def dfs(graph, node, visited):

if node not in visited:

print(node, end=' ')

visited.add(node)

for neighbor in graph[node]:

dfs(graph, neighbor, visited)

# BFS - Iterative using queue

def bfs(graph, start):

visited = set()

queue = deque([start])

visited.add(start)

while queue:

node = queue.popleft()

print(node, end=' ')

for neighbor in graph[node]:

if neighbor not in visited:

visited.add(neighbor)

queue.append(neighbor)

# Driver code

print("DFS Traversal:")

dfs(graph, '1', set())

print("\nBFS Traversal:")

bfs(graph, '1')

**OUTPUT:**

/home/student/PycharmProjects/PythonProject/.venv/bin/python /home/student/PycharmProjects/PythonProject/AI.py

DFS Traversal:

1 2 4 3 5 6 7

BFS Traversal:

1 2 3 4 5 6 7

Process finished with exit code 0

