### Experiment – No-3

### *Problem Statement :* Breadth-First Search (BFS) is an algorithm used for traversing graphs or trees. Traversing means visiting each node of the graph. Breadth-First Search is a recursive algorithm to search all the vertices of a graph or a tree. BFS in python can be implemented by using data structures like a dictionary and lists. Breadth-First Search in tree and graph is almost the same. The only difference is that the graph may contain cycles, so we may traverse to the same node again.

### C:\Users\Akshay singh\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot (160).png

***Program:***

graph = {

5 : [ 3 , 7 ] ,

3 : [ 2 , 4 ] ,

7 : [ 8 ] ,

2 : [ ] ,

4 : [ 8 ] ,

8 : [ ]

}

visited = []

queue = []

def Bfs(visited, graph, node):

visited.append(node)

queue.append(node)

while queue:

s = queue.pop(0)

print(s, end=" ")

for neighbour in graph[s]:

if neighbour not in visited:

visited.append(neighbour)

queue.append(neighbour)

print("The Traversal is:")

Bfs(visited, graph, 5)

### *Output :*

### *C:\Users\Akshay singh\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot (163).png*