Date: 29-07-25

Task - 1

Implementation of Graph Search algorithms (Breadth first Search and Depth First search) using following contraints

1a.

Aim: To implement of Graph Search algorithms (Breadth first Search and Depth first search) using python.

Algorithm:

BFS

1. Start by putting any one of the graph's vertices at the back of the queue
2. Now take the front item of the queue and add it to the visited list
3. Create a list of that vertex's adjacent nodes. Add those which are not within the visited list to the rear of the queue
4. Keep continuing steps two and three till the queue is empty.

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:
```

Output:

Following is the Breadth First Search

5 3 7 2 4 8

```
m = queue.pop(0)
print(m, end = " ")
for neighbour in graph[m]:
    if neighbour not in visited:
        visited.append(neighbour)
        queue.append(neighbour)
print("Following is the Breadth-first Search")
bfs(visited, graph, '5')
```

Task 1b.

(Algorithm).

Aim: To implement of Graph search algorithms using python

Algorithm:

DFS.

1. Declare a queue and insert the starting vertex.
2. Initialize a visited array and mark the starting vertex as visited
3. Remove the first vertex of queue.
4. Mark that vertex as visited.
5. Insert all the unvisited neighbours of the vertex into queue.
6. Stop.

Program:

```
graph = {
  '5' : ['3', '7'],
  '3' : ['2','4'],
  '7' : '8',
  '2' : [],
  '4' : ['8'],
  '8' : []
}
```

Output:

Executing DFS on depth-first search

5
3
2
4
8
7

```
visited = set()
def dfs(visited, graph, node):
    if node not in visited:
        print(node)
        visited.add(node)
        for neighbour in graph[node]:
            dfs(visited, graph, neighbour)
print("Following is the Depth-First search")
dfs(visited, graph, '5').
```

M. TECH - CSE	
...G.	
...FORMANCE (5)	
...SULT AND ANALYSIS (5)	
VIVA VOCE (5)	
GITHUB ...D (5)	
...L (20)	
...WITH DATE	

Result: Thus the implementation of Graph search algorithms (BFS and DFS) using python was successfully executed and output was verified