LP Practical -1

BFS Program –

```
graph = {
    'A' :['B','C'],
    'C' :['F'],
    'H' :['D'],
    'E' :['F','H'],
    'F' :[],
    'B' :['H','E'],
    'D' :[],
}

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)

 #driver code


bfs(visited, graph, 'A')
```

ouput -

```
[Running] python -u "c:\Users\STUDENT\Desktop\moksha\BFS.py"
ABCHEFD
[Done] exited with code=0 in 0.094 seconds
```

DFS Program –

```
graph = {
    'A' :['B','C'],
    'C' :['F'],
    'H' :['D'],
    'E' :['F','H'],
    'F' :[],
    'B' :['H','E'],
    'D' :[],
}

visited = set() # Set to keep track of visited nodes of graph.
def dfs(visited, graph, node): #function for dfs

    if node not in visited:
        print (node)
    visited.add(node)
    for neighbour in graph[node]:
        dfs(visited, graph, neighbour)

# Driver Code
print("Following is the Path using Depth-First Search")
dfs(visited, graph, 'A')
```

ouput -