T.Y. B.Sc. C.S. Sem-V	Roll No: **713**
	Date:31/08/2020

Practical no 2

AIM: Implement Iterative deep depth first search for Romanian map problem or any other map

CODE

```
from collections import defaultdict
class Graph:
    def __init__(self):
        self.graph = defaultdict(list)
    def addEdge(self, u, v):
        self.graph[u].append(v)
    def DFSUtil(self, v, visited):
        visited[v] = True
        print(v, end = ' ')
        for i in self.graph[v]:
            if visited[i] == False:
                self.DFSUtil(i, visited)
    def DFS(self, v):
        visited = [False] * (max(self.graph)+1)
        self.DFSUtil(v, visited)
g = Graph()
g.addEdge(0, 1)
g.addEdge(0, 2)
g.addEdge(1, 2)
g.addEdge(2, 0)
g.addEdge(2, 3)
```

```
g.addEdge(3, 3)
g.addEdge(3, 4)
g.addEdge(4, 4)
g.addEdge(4, 5)
g.addEdge(5, 4)
g.addEdge(5, 5)
g.addEdge(4, 6)
g.addEdge(5, 6)
g.addEdge(6, 6)

print("Following is DFS from (starting from vertex 0)")
print("Performed By krunal 713")
g.DFS(0)
```

```
================ RESTART: C:/Users/BlackBot/Desktop/AI_prac2.py ================
Following is DFS from (starting from vertex 0)
Performed By krunal 713
0 1 2 3 4 5 6
>>>
```