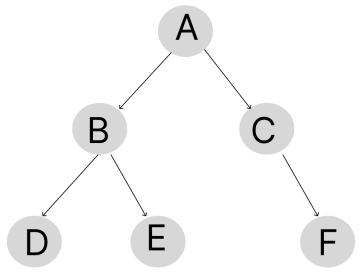
IMPLEMENTATION OF DFS

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

In [ ]:
```
from IPython.display import Image
Image(filename="Graph01.png" ,width=200,height=200)
```

Out[ ]:



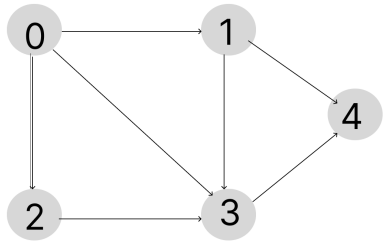
In [ ]:
```
def dfs_iterative(graph, start):
    visited = set()
    stack = [start]
    result = []

    while stack:
        node = stack.pop()
        if node not in visited:
            visited.add(node)
            result.append(node)
            stack.extend(graph[node][::-1])

    return result
# Test the functions
print("DFS (Iterative)", dfs_iterative(graph, 'A'))
```

```
DFS (Iterative) ['A', 'B', 'D', 'E', 'C', 'F']
```

In [ ]:
```
Graph = {
    0: [1, 2, 3],
    1: [4, 3],
    2: [3],
    3: [4],
    4: []
}
```

In [ ]:
```
from IPython.display import Image
Image(filename="Graph02.png" ,width=200,height=200)
```

Out[ ]:



In [ ]:
```python
def dfs_iterative(Graph, start):
    visited = set()
    stack = [start]
    result = []

    while stack:
        node = stack.pop()
        if node not in visited:
            visited.add(node)
            result.append(node)
            if node in Graph:
                for neighbor in reversed(Graph[node]):
                    stack.append(neighbor)

    return result


print("DFS (Iterative):", dfs_iterative(Graph, 0))
```

```
DFS (Iterative): [0, 1, 4, 3, 2]
```

AI LAB :[PC-CS(AM)593]

SUPRATIM NAG_AIML/22/57