

Name: Sanket Kanase AssingmentNo:01  
Div:B Batch:T5 RollNo:TCOB45

**BFS:-**

```
graph = {
    'S' : ['A', 'D'],
    'A' : ['C'],
    'D' : ['G'],
    'C' : ['G'],
    'G' : []
}

def BFS_goal(graph:dict,start,goal)->list[str]:
    queue = []
    res = []
    queue.append(start)
    while queue:
        elem = queue.pop(0)
        if elem not in res:

            res.append(elem)
            if elem==goal:return res

            for vertex in graph[elem]:
                queue.append(vertex)

    return goal + " Node Not Found"
for element in BFS_goal(graph,"S","C"):
    print(element,end=" ")
print("")
```

Name: Sanket Kanase AssingmentNo:01  
Div:B Batch:T5 RollNo:TCOB45

### DFS :-

```
graph = {
    'S': ['A', 'D'],
    'A': ['C'],
    'D': ['G'],
    'C': ['G'],
    'G': []
}

def DFS_goal(graph: dict, start: str, goal: str) -> list[str]:
    visited = set()
    res = []

    def dfs_helper(node: str):
        if node in visited:
            return None
        visited.add(node)
        res.append(node)
        if node == goal:
            return res
        for vertex in graph.get(node, []):
            path = dfs_helper(vertex)
            if path:
                return path
        return None

    result = dfs_helper(start)
    if result:
        return result
    return goal + " Node Not Found"

for element in DFS_goal(graph, "S", "C"):
    print(element, end=" ")
print("")
```