```
def DFS(graph, start, visited = None):
  if visited is None:
    visited = set()

  order = []

  if start not in visited:
    order.append(start)
    visited.add(start)
    print('------------------------------')
    print(f"Order: { order }")
    print(f"Visited : { visited }")

    for node in graph[start]:
      if node not in visited:
        order.extend(DFS(graph, node, visited))

    return order

graph = { 'A':['B','C'] , 'B':['A','D','E'] , 'C':['A','F','G'] , 'D':['B'] , 'E':['B'] , 'F':['C'] , 'G':['C'] }
DFS(graph, 'A')
```

OUTPUT

```
------------------------------
Order: ['A']
Visited : {'A'}
------------------------------
Order: ['B']
Visited : {'B', 'A'}
------------------------------
Order: ['D']
Visited : {'D', 'B', 'A'}
------------------------------
Order: ['E']
Visited : {'D', 'B', 'A', 'E'}
------------------------------
Order: ['C']
Visited : {'B', 'A', 'C', 'D', 'E'}
------------------------------
Order: ['F']
Visited : {'B', 'A', 'C', 'F', 'D', 'E'}
------------------------------
Order: ['G']
Visited : {'G', 'B', 'A', 'C', 'F', 'D', 'E'}
['A', 'B', 'D', 'E', 'C', 'F', 'G']
```

```
def DFS(graph, start, visited = None):
  if visited is None:
    visited = set()

  order = []

  if start not in visited:
    order.append(start)
    visited.add(start)

    for node in graph[start]:
      if node not in visited:
        order.extend(DFS(graph, node, visited))

    return order

graph = { 'A':['B','C'] , 'B':['A','D','E'] , 'C':['A','F','G'] , 'D':['B'] , 'E':['B'] , 'F':['C'] , 'G':['C'] }
DFS(graph, 'A')
```

OUTPUT

```
['A', 'B', 'D', 'E', 'C', 'F', 'G']
```