Depth First Search

Aim:

To implement DFS algorithm to traverse a graph

Algorithm

1. Start
2. Initialize a set to keep track of visited nodes
3. Start at starting node
4. Mark nodes as visited
5. Process current node
6. Apply DFS recursively if neighbor is not visited.
7. Stop

```
def dfs(graph, start, visited = None):
    if visited is None:
        visited = set()
    visited.add(start)
    print(start, end = ' ')
    for neighbour in graph[start]:
        if neighbour not in visited:
            dfs(graph, neighbor, visited)
    return visited

if __name__ = "__main__":
    graph = {
        'A':['B', 'C']
        'B':['A', 'D', 'E'],
        'C':['A', 'F']
        'D' : ['B']
        'E':['B', 'F']
        'F': ['C', 'E']
    }
    dfs(graph, 'A')
```

Output:

```
A B D E F C
```

Result:

The program has been executed successfully and the output has been verified