EXP NO : 2 - DFS

AIM:

To implement DFS Algorithm to traverse a graph.

ALGORITHM :

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

CODE:

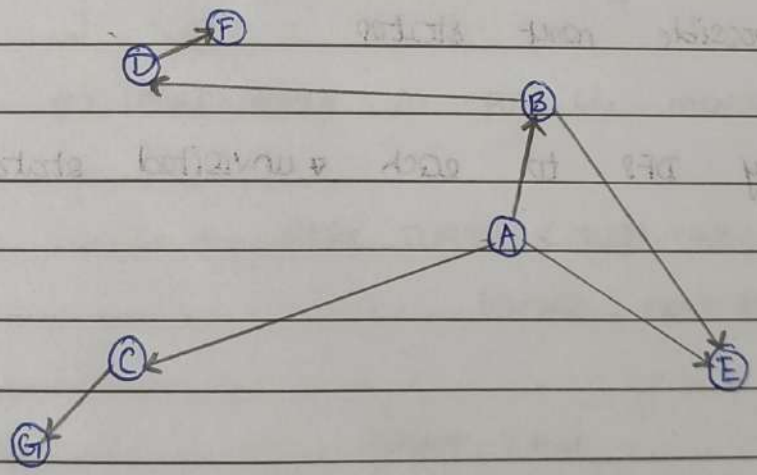
```
def DFS (graph , start, visited = none):
    if visited is none :
        visited = set ()
    visited. add (start)
    print (start, end =" ")
    for neighbour in graph[start]:
        if neighbour non in visited :
            DFS (graph , neighbour, visited)
    return visited.
if __name__ = "__main__"
```

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

OUTPUT:

Following is DFS from (starting from vertex A)

ABDFECG



RESULT:

Thus the program to implement DFS algorithm is successfully executed and output is verified.