Aim:

To write a program to achieve a graph traversal algorithm using DFS

Algorithm:

Step 1: Start

Step 2: Initialize a Stack (or use recursion) to keep track of nodes to visit.

Step 3: Start from the chosen node (root) and mark it as visited

Step 4: Push the starting node onto the Stack

Step 5: While the Stack is not empty pop the node from the stack mark as visited push the neighbour onto the Stack

Step 6: Repeat until all nodes have been visited or stack is empty

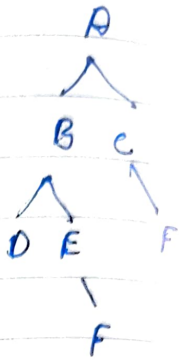
Step 7: End.

Program:

```
graph = {
'A' : ['B', 'C'],
'B' : ['D', 'E'],
'C' : ['F'],
'D' : []
'E' : [F]
'F' : []
}
Visited = set()
def dfs (graph, node)
    if node not in visited
        Print (node, end = " ")
        visited. add (node)
        for neighbour in graph [node]:
            dfs (graph, neighbour)
start_node = Input ("Enter the start node:")
dfs (graph, start_node)
```

Output:

Enter the start node: A
A B D E F C



Result:

Thus the program for the DFS has executed Successfully.