```
#program to demonstrate Depth First Search

from collections import defaultdict
class Graph:
    def __init__(self):
        self.graph=defaultdict(list)
    def addEdge(self,u,v):
        self.graph[u].append(v)
    def DFSutil(self,v,visited):
        visited.add(v)
        print(v,end=' ')

        for neighbour in self.graph[v]:
            if neighbour not in visited:
                self.DFSutil(neighbour,visited)

    def DFS(self,v):
        visited=set()
        self.DFSutil(v,visited)

g=Graph()
g.addEdge('0','1')
g.addEdge('0','3')
g.addEdge('1','2')
g.addEdge('1','3')
g.addEdge('2','4')
g.addEdge('2','1')
g.addEdge('4','2')
g.addEdge('4','3')
g.addEdge('3','1')
g.addEdge('3','4')
source=input('enter a String vertex:')
print("Following is Depth First Traversal" "(starting From
vertex)",source);
g.DFS(source)

*******************************OUTPUT***********************************
enter a String vertex:0
Following is Depth First Traversal(starting From vertex) 0
0 1 2 4 3
```