

```

1
2 def dfs(graph, start, visited=None):
3     if visited is None:
4         visited = set() # To track visited vertices
5
6     visited.add(start) # Mark the start vertex as visited
7     print(start, end=' ') # Print the visited vertex
8
9     # Recur for all the vertices adjacent to the current vertex
10    for neighbor in graph[start]:
11        if neighbor not in visited:
12            dfs(graph, neighbor, visited)
13
14    return visited
15
16 # Example graph represented as an adjacency list
17 graph = {
18     'A': ['B', 'C'],
19     'B': ['A', 'D', 'E'],
20     'C': ['A', 'F'],
21     'D': ['B'],
22     'E': ['B', 'F'],
23     'F': ['C', 'E']
24 }
25
26 # Starting DFS from vertex 'A'
27 dfs(graph, 'A')

```



A B D E F C

...Program finished with exit code 0
Press ENTER to exit console.