DFS 1

```
import itertools


def dfs(graph, start):
    visited = set()
    stack = [start]


    while stack:
        vertex = stack.pop()
        if vertex not in visited:
            visited.add(vertex)
            print(vertex)  # Replace with desired operation on the vertex


            neighbors = graph[vertex]
            unvisited_neighbors = itertools.filterfalse(visited.__contains__, neighbors)
            stack.extend(reversed(list(unvisited_neighbors)))


# User input for constructing the graph
graph = {}
n = int(input("Enter the number of vertices in the graph : "))
for i in range(n):
    vertex = input(f"Enter vertex {i + 1} : ")
    neighbors = input(f"Enter neighbors of vertex {i + 1} (space-separated) : ").split()
    graph[vertex] = neighbors


start_vertex = input("Enter the starting vertex for DFS: ")


# Calling the DFS function with user-provided inputs
dfs(graph, start_vertex)
```