```
From collections import deque

Def bfs(graph,start):
    Visited=set()
    Queue=deque([start])

    While queue:
        Node=queue.popleft()
        If node not in visited:
            Visited.add(node)
            Print(node)
            For neighbor in graph[node]:
                If neighbor not in visited:
                    Queue.append(neighbor)

Def dfs(graph,start,visited=None):
    If visited is None:
        Visited=set()
    Visited.add(start)
    Print(start)
    For neighbor in graph[start]:
        If neighbor not in visited:
            Dfs(graph,neighbor,visited)

Graph={}

While True:
    Vertex=input(“enter vertex(‘done’ if finished)”)
    If vertex==’done’:
        Break
    Neighbors=input(“neighbor for vertex{vertex}:”).split()
    Graph[vertex]=neighbors

Print(“bfs”)
Bfs(graph,’A’)

Print(“dfs”)
Dfs(graph,’A’)
```