BFS 2

```
import collections


# Function to perform Breadth First Search
def bfs(graph, start, goal):
    visited = set()  # Set to keep track of visited vertices
    queue = collections.deque([start])  # Queue for BFS traversal
    visited.add(start)  # Mark the start vertex as visited

    while queue:
        vertex = queue.popleft()
        print(vertex)

        # Check if the current vertex is the goal
        if vertex == goal:
            print("GOAL FOUND")
            return

        # Visit all the adjacent vertices of the current vertex
        for neighbor in graph[vertex]:
            if neighbor not in visited:
                queue.append(neighbor)
                visited.add(neighbor)

# Main program
if __name__ == "__main__":
    graph = collections.defaultdict(list)

    # Get the number of nodes from the user
    num_nodes = int(input("Enter the number of nodes: "))
```

```
# Construct the graph
for i in range(1, num_nodes + 1):
    node = input("Enter node {}: ".format(i))
    adj_nodes = int(input("Enter the number of adjacent nodes: "))
    for j in range(adj_nodes):
        adj_node = input("Enter adjacent node: ")
        graph[node].append(adj_node)

start_node = input("Enter the starting node: ")
goal_node = input("Enter the goal node: ")

# Perform BFS
print("BFS traversal:")
bfs(graph, start_node, goal_node)
```