```
from collections import deque

def dfs(graph, node, visited):
    visited.add(node)
    print(node, end=' ')
    for neighbor in graph.get(node, []):
        if neighbor not in visited:
            dfs(graph, neighbor, visited)

def bfs(graph, start):
    visited = set([start])
    queue = deque([start])

    while queue:
        node = queue.popleft()
        print(node, end=' ')
        for neighbor in graph.get(node, []):
            if neighbor not in visited:
                visited.add(neighbor)
                queue.append(neighbor)

def create_graph():
    edges = int(input("\nEnter number of edges: "))
    graph = {}
    for _ in range(edges):
        u, v = input("Enter edge (u v): ").split()
        if u not in graph:
            graph[u] = []
        if v not in graph:
            graph[v] = []
        graph[u].append(v)
        graph[v].append(u)
    return graph

def main():
    graph = {}
    while True:
        print("\n=== Graph Traversal Menu ===")
        print("1. Create Graph")
        print("2. Perform DFS Traversal")
        print("3. Perform BFS Traversal")
        print("4. Exit")

        choice = input("Enter your choice: ")
        if choice == '1':
            graph = create_graph()
            print("Graph created successfully!")

        elif choice == '2':
            if graph:
                start = input("Enter start vertex for DFS: ")
                if start in graph:
                    print("DFS traversal:")
                    dfs(graph, start, set())
                    print()
                else:
                    print("Start vertex not found in graph.")
            else:
                print("Graph is empty. Please create it first.")

        elif choice == '3':
            if graph:
                start = input("Enter start vertex for BFS: ")
                if start in graph:
                    print("BFS traversal:")
                    bfs(graph, start)
                    print()
                else:
                    print("Start vertex not found in graph.")
            else:
                print("Graph is empty. Please create it first.")

        elif choice == '4':
            print("Exiting program successfully.")
            break

        else:
            print("Invalid choice! Please try again.")

if __name__ == "__main__":
    main()
```