Assignment 1 : BFS and DFS

```
class Graph:
    def __init__(self):
        self.graph = {}

    def add_edge(self, vertex, edge):
        if vertex not in self.graph:
            self.graph[vertex] = []
        self.graph[vertex].append(edge)

    def dfs(self, start_vertex):
        visited = set()

        def dfs_recursive(vertex):
            visited.add(vertex)
            print(vertex, end=" ")

            for neighbor in self.graph.get(vertex, []):
                if neighbor not in visited:
                    dfs_recursive(neighbor)

        dfs_recursive(start_vertex)
        print()

    def bfs(self, start_vertex):
        visited = set()
        queue = []

        visited.add(start_vertex)
        queue.append(start_vertex)

        while queue:
            vertex = queue.pop(0)
            print(vertex, end=" ")

            for neighbor in self.graph.get(vertex, []):
                if neighbor not in visited:
                    visited.add(neighbor)
                    queue.append(neighbor)

        print()

# Create an empty graph
```

```
g = Graph()

# Take user input for vertices and edges
while True:
    vertex = input("Enter a vertex (or 'done' to finish adding vertices): ")
    if vertex.lower() == 'done':
        break

    edge = input("Enter an edge for {}: ".format(vertex))
    g.add_edge(vertex, edge)

# Take user input for traversal type
while True:
    traversal_type = input("Enter 'DFS' or 'BFS' to perform traversal (or 'exit' to quit): ").upper()

    if traversal_type == 'EXIT':
        break
    elif traversal_type == 'DFS':
        start_vertex = input("Enter the starting vertex: ")
        print("DFS traversal:")
        g.dfs(start_vertex)
    elif traversal_type == 'BFS':
        start_vertex = input("Enter the starting vertex: ")
        print("BFS traversal:")
        g.bfs(start_vertex)
    else:
        print("Invalid input. Please enter 'DFS' or 'BFS' (or 'exit' to quit).")
```

```
Enter a vertex (or 'done' to finish adding vertices): 1
Enter an edge for 1: 2
Enter a vertex (or 'done' to finish adding vertices): 1
Enter an edge for 1: 3
Enter a vertex (or 'done' to finish adding vertices): 2
Enter an edge for 2: 6
Enter a vertex (or 'done' to finish adding vertices): 6
Enter an edge for 6: 4
Enter a vertex (or 'done' to finish adding vertices): 4
Enter an edge for 4: 3
Enter a vertex (or 'done' to finish adding vertices): done
Enter 'DFS' or 'BFS' to perform traversal (or 'exit' to quit): dfs
Enter the starting vertex: 1
DFS traversal:
1 2 6 4 3
Enter 'DFS' or 'BFS' to perform traversal (or 'exit' to quit):
```