Experiment 8: BREADTH FIRST SEARCH

AIM: Implement an Algorithm in Python to solve the BFS problem.

Code:

```
from collections import deque


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

    while queue:
        current, path = queue.popleft()

        if current == goal:
            return path

        for neighbor in graph[current]:
            if neighbor not in visited:
                visited.add(neighbor)
                queue.append((neighbor, path + [neighbor]))

    return None

# Example usage
graph = {
    'A': ['B', 'C'],
    'B': ['A', 'D', 'E'],
    'C': ['A', 'F'],
```

```
    'D': ['B'],
    'E': ['B', 'F'],
    'F': ['C', 'E']
}

start_node = 'A'
goal_node = 'F'

path = bfs(graph, start_node, goal_node)
print("Path:", path)
```

Output:

Path: ['A', 'C', 'F']

Result: Code has been Implemented successfully.