```python
# Experiment 1: Implementation of Uninformed Search Strategies:
from collections import deque

graph = {
    'A': ['B', 'C'],
    'B': ['A', 'D', 'E'],
    'C': ['A', 'F'],
    'D': ['B'],
    'E': ['B', 'F'],
    'F': ['C', 'E']
}

def dfs(graph, start, goal, path=[]):
    path = path + [start]
    if start == goal:
        return path
    for node in graph[start]:
        if node not in path:
            newpath = dfs(graph, node, goal, path)
            if newpath:
                return newpath
    return None

def bfs(graph, start, goal):
    visited = set()
    queue = deque([[start]])
    while queue:
        path = queue.popleft()
        node = path[-1]
        if node == goal:
            return path
        if node not in visited:
            for neighbor in graph[node]:
                new_path = list(path)
                new_path.append(neighbor)
                queue.append(new_path)
            visited.add(node)
    return None

start = input("Enter start node: ").upper()
goal = input("Enter goal node: ").upper()

print("DFS Path:", dfs(graph, start, goal))
print("BFS Path:", bfs(graph, start, goal))
```

```
PS C:\Users\ROG STRIX\Downloads\omkar> & "C:/Users/ROG STRIX/AppData/Local/Microsoft/WindowsApps/python3.13.exe" "c:/Users/ROG STRIX/Downloads/python/from collections import deque.py"
Enter start node: A
Enter goal node: F
DFS Path: ['A', 'B', 'E', 'F']
BFS Path: ['A', 'C', 'F']
PS C:\Users\ROG STRIX\Downloads\omkar>
```