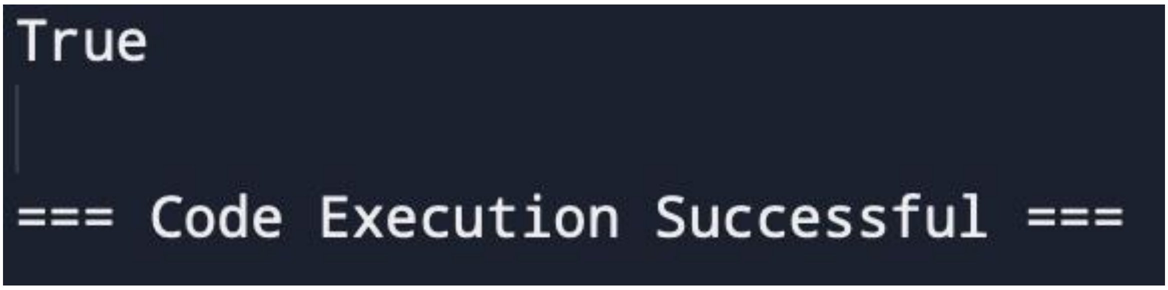
117. Hamiltonian Cycle Problem

PROGRAM:-

```
def hamiltonian_cycle(graph, start_v):
    def hamiltonian_util(v, visited):
        if len(visited) == len(graph):
            return start_v in graph[v]

        for next_v in graph[v]:
            if next_v not in visited:
                visited.add(next_v)
                if hamiltonian_util(next_v, visited):
                    return True
                visited.remove(next_v)

        return False

    visited = set([start_v])
    return hamiltonian_util(start_v, visited)

# Example Usage
graph = {
    'A': ['B', 'C', 'D'],
    'B': ['A', 'C', 'D'],
    'C': ['A', 'B', 'D'],
    'D': ['A', 'B', 'C']
}

start_vertex = 'A'
print(hamiltonia
n_cycle(graph, start_vertex))  # Output: True
```

OUTPUT:-



```
True

=== Code Execution Successful ===
```

TIME COMPLEXITY:-O(n!)