

### 117. Hamiltonian Cycle Problem

PROGRAM:-

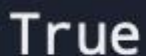
```
def hamiltonian_cycle(graph, start_v):  
    def hamiltonian_util(v, visited):  
        if len(visited) == len(graph):  
            return start_v in graph[v]  
  
        for next_v in graph[v]:  
            if next_v not in visited:  
                visited.add(next_v)  
                if 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(hamiltonian_cycle(graph, start_vertex)) # Output: True
```

OUTPUT:-

A dark-themed terminal window showing the output of the program. The word "True" is displayed in a large, white, monospace font.A dark-themed terminal window showing a success message. The text "=== Code Execution Successful ===" is displayed in a white, monospace font, centered on the line.

TIME COMPLEXITY:- $O(n!)$