

# Practical 7

Name :Virendra A Uplenchwar

Roll\_no:A4\_B1\_05

## CODE:-

```
def safe(i, graph, path, pos):
    if graph[path[pos - 1]][i] == 0:
        return False
    if i in path:
        return False
    return True

def solve(graph, path, pos):
    if pos == len(graph):
        if graph[path[pos - 1]][path[0]] == 1:
            return True
        else:
            return False
    for i in range(1, len(graph)):
        if safe(i, graph, path, pos):
            path[pos] = i
            if solve(graph, path, pos + 1):
                return True
            path[pos] = -1
    return False

def hamiltonian(graph, labels):
    path = [-1] * len(graph)
    path[0] = 0
    if not solve(graph, path, 1):
        print("No Hamiltonian Cycle exists")
    else:
        path.append(path[0])
        print("Hamiltonian Cycle:", [labels[i] for i in path])

def main():
```

```
print("Choose a graph to find the Hamiltonian Cycle:")
print("1. Graph with vertices A, B, C, D, E")
print("2. Graph with vertices T, M, S, H, C")

choice = int(input("Enter your choice (1 or 2): "))

if choice == 1:
    graph = [
        [0, 1, 1, 0, 1],
        [1, 0, 1, 1, 0],
        [1, 1, 0, 1, 0],
        [0, 1, 1, 0, 1],
        [1, 0, 0, 1, 0]
    ]
    labels = ['A', 'B', 'C', 'D', 'E']
    hamiltonian(graph, labels)

elif choice == 2:
    graph = [
        [0, 1, 1, 0, 1],
        [1, 0, 1, 1, 0],
        [1, 1, 0, 1, 1],
        [0, 1, 1, 0, 1],
        [1, 0, 1, 1, 0]
    ]
    labels = ['T', 'M', 'S', 'H', 'C']
    hamiltonian(graph, labels)

else:
    print("Invalid choice")

main()
```

## OUTPUT:-

```
Choose a graph to find the Hamiltonian Cycle:  
1. Graph with vertices A, B, C, D, E  
2. Graph with vertices T, M, S, H, C  
Enter your choice (1 or 2): 1  
Hamiltonian Cycle: ['A', 'B', 'C', 'D', 'E', 'A']
```

```
Choose a graph to find the Hamiltonian Cycle:  
1. Graph with vertices A, B, C, D, E  
2. Graph with vertices T, M, S, H, C  
Enter your choice (1 or 2): 2  
Hamiltonian Cycle: ['T', 'M', 'S', 'H', 'C', 'T']
```