Assignment No. 15

Name : Manish Namdev Barage

PRN : 22520007

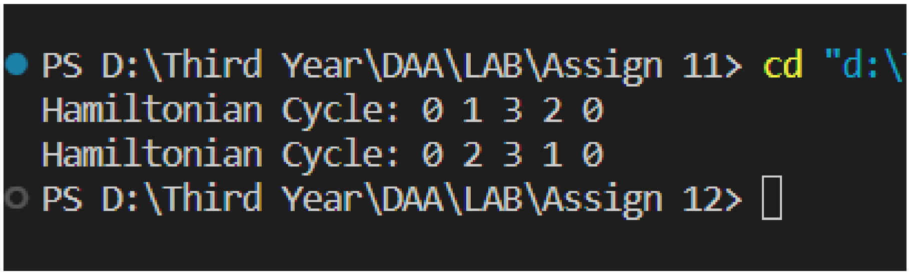
Batch : T7

Topic : To implement Hamiltonian cycle using backtracking

Code :

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

void printHamiltonianCyclesUtil(vector<vector<int>>& graph,
vector<int>& path, vector<bool>& visited, int pos) {
    if (pos == graph.size()) {
        if (find(graph[path[pos - 1]].begin(), graph[path[pos -
1]].end(), path[0]) != graph[path[pos - 1]].end()) {
            // Print the Hamiltonian cycle
            cout << "Hamiltonian Cycle: ";
            for (int vertex : path) {
                cout << vertex << " ";
            }
            cout << path[0] << endl; // Cycle back to the first vertex
        }
        return;
    }

    for (int v : graph[path[pos - 1]]) {
        if (!visited[v]) {
            visited[v] = true;
            path[pos] = v;
            printHamiltonianCyclesUtil(graph, path, visited, pos + 1);
            visited[v] = false; // Backtrack
            path[pos] = -1;
        }
    }
}

void printHamiltonianCycles(vector<vector<int>>& graph) {
    vector<int> path(graph.size(), -1);
    vector<bool> visited(graph.size(), false);

    path[0] = 0; // Start from the first vertex
    visited[0] = true;

    printHamiltonianCyclesUtil(graph, path, visited, 1);
}

int main() {
    vector<vector<int>> graph = {
        {1, 2},
        {0, 2, 3},
        {0, 1, 3},
        {1, 2}
    };

    printHamiltonianCycles(graph);
    return 0;
}
```

Output:



```
PS D:\Third Year\DAA\LAB\Assign 11> cd "d:\
Hamiltonian Cycle: 0 1 3 2 0
Hamiltonian Cycle: 0 2 3 1 0
PS D:\Third Year\DAA\LAB\Assign 12>
```