1. **Hamiltonian circuit**

**Code:**

#include <stdio.h>

#include <stdbool.h>

#define MAX\_VERTICES 10

// Function to check if the vertex v can be added at index 'pos' in the Hamiltonian cycle

bool isSafe(int v, bool graph[][MAX\_VERTICES], int path[], int pos, int V) {

// Check if this vertex is adjacent to the previously added vertex and not already added

if (!graph[path[pos - 1]][v])

return false;

// Check if the vertex has already been included

for (int i = 0; i < pos; i++)

if (path[i] == v)

return false;

return true;

}

// Recursive utility function to find a Hamiltonian circuit starting from the vertex 'pos'

bool hamiltonianCircuitUtil(bool graph[][MAX\_VERTICES], int path[], int pos, int V) {

// If all vertices are included in the Hamiltonian cycle

if (pos == V) {

// Check if there is an edge from the last vertex to the first vertex

if (graph[path[pos - 1]][path[0]])

return true;

else

return false;

}

// Try different vertices as the next candidate in Hamiltonian cycle

for (int v = 1; v < V; v++) {

if (isSafe(v, graph, path, pos, V)) {

path[pos] = v;

// Recur to construct the rest of the path

if (hamiltonianCircuitUtil(graph, path, pos + 1, V))

return true;

// If adding vertex v doesn't lead to a solution, remove it

path[pos] = -1;

}

}

return false;

}

// Function to find and print the Hamiltonian circuit in the given graph

void hamiltonianCircuit(bool graph[][MAX\_VERTICES], int V) {

int path[MAX\_VERTICES];

// Initialize all vertices as not included in the Hamiltonian cycle

for (int i = 0; i < V; i++)

path[i] = -1;

// Start from vertex 0 as the first vertex in the path

path[0] = 0;

if (!hamiltonianCircuitUtil(graph, path, 1, V)) {

printf("Hamiltonian circuit does not exist\n");

return;

}

printf("Hamiltonian circuit exists: ");

for (int i = 0; i < V; i++)

printf("%d ", path[i]);

printf("%d\n", path[0]);

}

// Main function

int main() {

int V;

printf("Enter the number of vertices in the graph: ");

scanf("%d", &V);

printf("Enter the adjacency matrix (0/1) for the graph:\n");

bool graph[MAX\_VERTICES][MAX\_VERTICES];

for (int i = 0; i < V; i++) {

for (int j = 0; j < V; j++) {

scanf("%d", &graph[i][j]);

}

}

hamiltonianCircuit(graph, V);

return 0;

}

**Output:**

Enter the number of vertices in the graph: 4

Enter the adjacency matrix (0/1) for the graph:

0 1 1 0

1 0 1 1

1 1 0 1

0 1 1 0

Hamiltonian circuit exists: 0 1 3 2 0

--------------------------------

Process exited after 42 seconds with return value 0

Press any key to continue . . .

