

Practical no: 7

Name: Saloni Vishwakarma

Batch-Roll no: C1-13

Subject: DAA Lab

Date of execution: 1 July 2023

Aim: To implement Hamiltonian Cycle algorithm using recursion in C.

Hamiltonian Cycle Algorithm (Code and Output):

```
#include <stdio.h>
#include <stdbool.h>

#define MAX 10

int graph[MAX][MAX];
int path[MAX];
int vertices;
int count = 0;

bool isSafe(int v, int k) {
    if (graph[path[k - 1]][v] == 0)
        return false;
    int i;
    for (i = 0; i < k; i++) {
        if (path[i] == v)
            return false;
    }

    return true;
}

void printHamiltonianCycle() {
    int i;
    printf("\n Hamiltonian Cycle %d:\n", ++count);

    for (i = 0; i < vertices; i++) {
```

```

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

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

void hamiltonianCycleUtil(int k) {
    if (k == vertices) {
        if (graph[path[k - 1]][path[0]] == 1)
            printHamiltonianCycle();
        return;
    }
    int v;
    for (v = 1; v <= vertices; v++) {
        if (isSafe(v, k)) {
            path[k] = v;
            hamiltonianCycleUtil(k + 1);
            path[k] = -1;
        }
    }
}

void hamiltonianCycle(int n) {
    int i;
    for (i = 0; i <= n; i++)
        path[i] = -1;

    path[0] = 1;
    hamiltonianCycleUtil(1);
}

int main() {
    printf("\n Enter the total number of vertices: ");
    scanf("%d", &vertices);
    int i,j;

    printf("\n Enter the adjacency matrix: \n");
    for (i = 1; i <= vertices; i++) {
        printf(" ");
        for (j = 1; j <= vertices; j++) {

```

```

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

hamiltonianCycle(vertices);

return 0;
}

```

```

Enter the total number of vertices: 6

Enter the adjacency matrix:
0 1 1 1 1 0
1 0 0 1 0 0
1 0 0 1 1 1
1 1 1 0 0 1
1 0 1 0 0 0
0 0 1 1 0 0

Hamiltonian Cycle 1:
1-> 2-> 4-> 6-> 3-> 5->1

Hamiltonian Cycle 2:
1-> 5-> 3-> 6-> 4-> 2->1

...Program finished with exit code 0
Press ENTER to exit console.

```

Conclusion: We have successfully studied and implemented Hamiltonian Cycle algorithm using recursion in C.