DAA

Name – Chetna Deshmukh

Section – A4

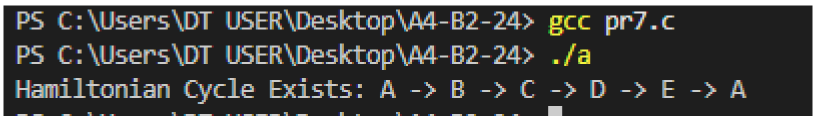
Roll No. – 24

Practical 7

Code:

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

#define V 5

void printSolution(int path[])
{
    int i;
    printf("Hamiltonian Cycle Exists: ");
    for (i = 0; i < V; i++)
        printf("%c -> ", path[i] + 'A');
    printf("%c\n", path[0] + 'A');
}

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

bool hamCycleUtil(int graph[V][V], int path[], int pos)
{
    int v;
    if (pos == V)
    {
        if (graph[path[pos - 1]][path[0]] == 1)
            return true;
        else
            return false;
    }
    for (v = 1; v < V; v++)
    {
        if (isSafe(v, graph, path, pos))
        {
            path[pos] = v;
            if (hamCycleUtil(graph, path, pos + 1))
                return true;
            path[pos] = -1;
        }
    }
    return false;
}

bool hamCycle(int graph[V][V])
{
    int path[V];
    int i;
    for (i = 0; i < V; i++)
        path[i] = -1;
    path[0] = 0;
    if (!hamCycleUtil(graph, path, 1))
    {
        printf("No Hamiltonian Cycle exists.\n");
        return false;
    }
    printSolution(path);
    return true;
}

int main()
{
    int graph[V][V] =
    {
        {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}
    };
    hamCycle(graph);
    return 0;
}
```

Output:



```
PS C:\Users\DT USER\Desktop\A4-B2-24> gcc pr7.c
PS C:\Users\DT USER\Desktop\A4-B2-24> ./a
Hamiltonian Cycle Exists: A -> B -> C -> D -> E -> A
```