DAA PRACTICAL 7

Name:Ansha Lanjewar
Section:A3_B1_13

Aim: Implement Hamiltonian Cycle using Backtracking.
Code:

```c
#include <stdio.h>
#define N 5

int G[N][N] = {
    {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}
};

int x[N];
int found = 0;
void printCycle() {
    for (int i = 0; i < N; i++)
        printf("%c ", x[i] + 'A');
    printf("%c\n", x[0] + 'A');
    found = 1;
}
void NextVertex(int k) {
    while (1) {
        x[k] = (x[k] + 1) % N;
        if (x[k] == 0)
            return;
        if (G[x[k - 1]][x[k]] != 0) {
            int j;
            for (j = 0; j < k; j++)
                if (x[j] == x[k])
                    break;
            if (j == k) {
```

```
                if (k < N - 1 || (k == N - 1 && G[x[k]][x[0]] != 0))
                    return;
            }
        }
    }
}

void Hamiltonian(int k) {
    while (1) {
        NextVertex(k);
        if (x[k] == 0)
            return;
        if (k == N - 1)
            printCycle();
        else
            Hamiltonian(k + 1);
    }
}
int main() {
    for (int i = 0; i < N; i++)
        x[i] = 0;
    x[0] = 0;
    Hamiltonian(1);
    if (found)
        printf("Hamiltonian cycle found.\n");
    else
        printf("No Hamiltonian cycle found.\n");

return 0;
}
```

OUTPUT:

```
[Running] cd "c:\Users\DT USER\Desktop\1A333333\DAA\A3
A B C D E A
A C B D E A
A E D B C A
A E D C B A
Hamiltonian cycle found.

[Done] exited with code=0 in 0.264 seconds
```

```
[Running] cd "c:\Users\DT USER\Desktop\1A333333\DAA\A333333\" && g++ DAA7.C -o DAA7 &
T U V W X T
T U W V X T
T U W X V T
T V U W X T
T V X W U T
T X V W U T
T X W U V T
T X W V U T
Hamiltonian cycle found.

[Done] exited with code=0 in 0.25 seconds
```