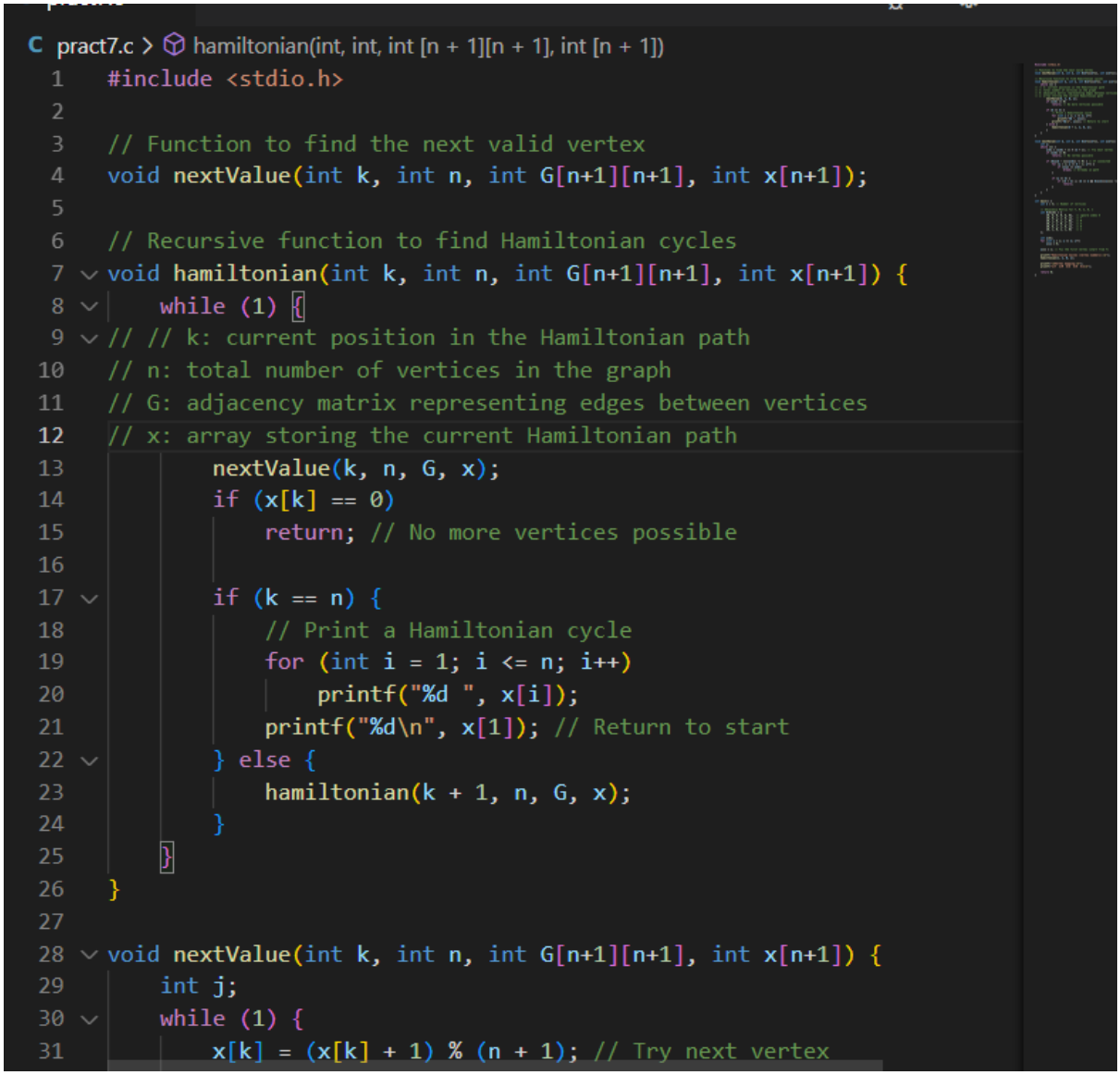
Name – Tanishq Mankari
Section - A8_B2
Roll number – 31

Practical – 7

Code-



```c
#include <stdio.h>

// Function to find the next valid vertex
void nextValue(int k, int n, int G[n+1][n+1], int x[n+1]);

// Recursive function to find Hamiltonian cycles
void hamiltonian(int k, int n, int G[n+1][n+1], int x[n+1]) {
    while (1) {
// // k: current position in the Hamiltonian path
// n: total number of vertices in the graph
// G: adjacency matrix representing edges between vertices
// x: array storing the current Hamiltonian path
        nextValue(k, n, G, x);
        if (x[k] == 0)
            return; // No more vertices possible

        if (k == n) {
            // Print a Hamiltonian cycle
            for (int i = 1; i <= n; i++)
                printf("%d ", x[i]);
            printf("%d\n", x[1]); // Return to start
        } else {
            hamiltonian(k + 1, n, G, x);
        }
    }
}

void nextValue(int k, int n, int G[n+1][n+1], int x[n+1]) {
    int j;
    while (1) {
        x[k] = (x[k] + 1) % (n + 1); // Try next vertex
```

```
        if (x[k] == 0)
            return; // No vertex possible

        if (G[x[k - 1]][x[k]] != 0) { // If connected
            for (j = 1; j <= k - 1; j++) {
                if (x[j] == x[k])
                    break; // Already in path
            }

            if (j == k) {
                if ((k < n) || (k == n && G[x[n]][x[1]] != 0))
                    return;
            }
        }
    }
}

int main() {
    int n = 5; // Number of vertices
```

```
    // Adjacency Matrix for T, M, S, H, C
    int G[6][6] = {
        {0, 0, 0, 0, 0, 0},  // ignore index 0
        {0, 0, 1, 1, 0, 1},  // T
        {0, 1, 0, 1, 1, 0},  // M
        {0, 1, 1, 0, 1, 1},  // S
        {0, 0, 1, 1, 0, 1},  // H
        {0, 1, 0, 1, 1, 0}   // C
    };

    int x[6];
    for (int i = 1; i <= n; i++)
        x[i] = 0;

    x[1] = 1; // Fix the first vertex (start from T)

    printf("Hamiltonian Cycles (vertex numbers):\n");
    hamiltonian(2, n, G, x);

    printf("\nVertex mapping:\n");
    printf("1=T  2=M  3=S  4=H  5=C\n");

    return 0;
}
```

Output –

```
Hamiltonian Cycles (vertex numbers):
1 2 3 4 5 1
1 2 4 3 5 1
1 2 4 5 3 1
1 3 2 4 5 1
1 3 5 4 2 1
1 5 3 4 2 1
1 5 4 2 3 1
1 5 4 3 2 1

Vertex mapping:
1=T  2=M  3=S  4=H  5=C
```