NAME: Haresh Kumar N L (192425009)

COURSE NAME: DATA STRUCTURES FOR MODERN COMPUTING SYSTEMS

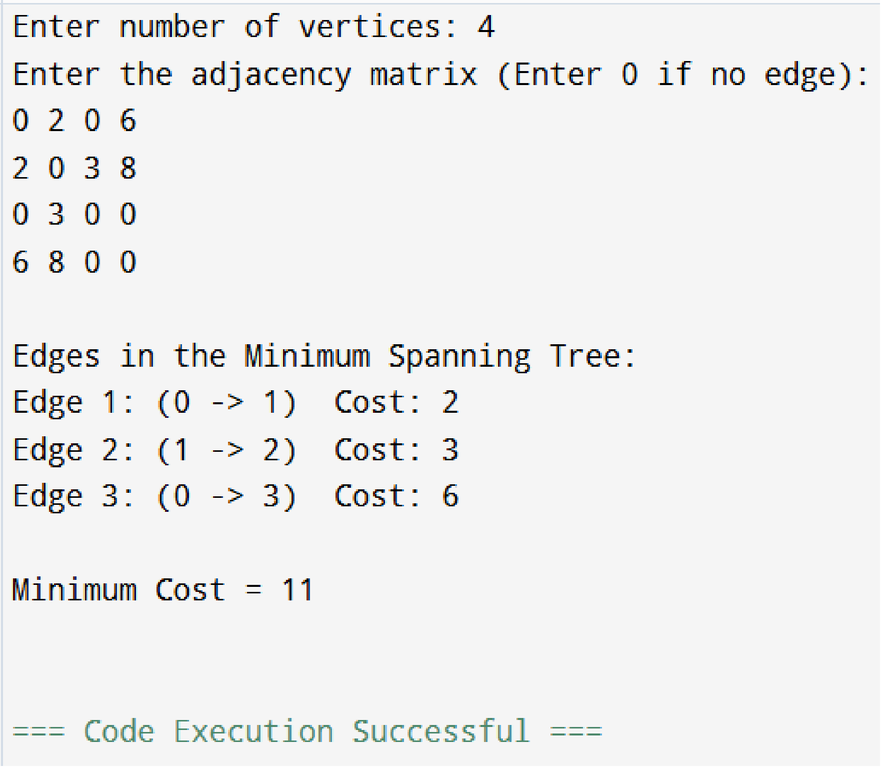
COURSE CODE: CSA0302

Experiment 37: Kruskal’s Algorithm

Code:

```
#include <stdio.h>
#define INF 9999
#define MAX 20
int parent[MAX];
int find(int i) {
    while (parent[i])
        i = parent[i];
    return i;
}
int union_set(int i, int j) {
    if (i != j) {
        parent[j] = i;
        return 1;
    }
    return 0;
}
int main() {
    int cost[MAX][MAX];
    int n;
    int min, a, b, u, v;
    int ne = 1, total = 0;
    printf("Enter number of vertices: ");
    scanf("%d", &n);
    printf("Enter the adjacency matrix (Enter 0 if no edge):\n");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &cost[i][j]);
            if (cost[i][j] == 0)
                cost[i][j] = INF;
        }
    }
    printf("\nEdges in the Minimum Spanning Tree:\n");
    while (ne < n) {
        min = INF;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                if (cost[i][j] < min) {
                    min = cost[i][j];
                    a = u = i;
                    b = v = j;
                }
            }
        }
        u = find(u);
        v = find(v);
        if (union_set(u, v)) {
            printf("Edge %d: (%d -> %d)  Cost: %d\n", ne++, a, b, min);
            total += min;
        }
        cost[a][b] = cost[b][a] = INF;
    }
    printf("\nMinimum Cost = %d\n", total);
    return 0;
}
```

Output:



```
Enter number of vertices: 4
Enter the adjacency matrix (Enter 0 if no edge):
0 2 0 6
2 0 3 8
0 3 0 0
6 8 0 0

Edges in the Minimum Spanning Tree:
Edge 1: (0 -> 1)  Cost: 2
Edge 2: (1 -> 2)  Cost: 3
Edge 3: (0 -> 3)  Cost: 6

Minimum Cost = 11


=== Code Execution Successful ===
```