

**NAME: Haresh Kumar N L (192425009)**

**COURSE NAME: DATA STRUCTURES FOR MODERN COMPUTING SYSTEMS**

**COURSE CODE: CSA0302**

Experiment 35: Prim's Algorithm

Code:

```
#include <stdio.h>

#define INF 9999

#define MAX 20

int main() {

    int cost[MAX][MAX];

    int visited[MAX] = {0};

    int n, i, j, ne = 1;

    int min, a, b, u, v, total = 0;

    printf("Enter number of vertices: ");

    scanf("%d", &n);

    printf("Enter the adjacency matrix (Enter 0 if no edge):\n");

    for (i = 0; i < n; i++) {

        for (j = 0; j < n; j++) {

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

            if (cost[i][j] == 0)

                cost[i][j] = INF;

        }

    }

    visited[0] = 1;

    printf("\nEdges in the Minimum Spanning Tree:\n");

    while (ne < n) {

        min = INF;

        for (i = 0; i < n; i++) {

            if (visited[i]) {

                for (j = 0; j < n; j++) {

                    if (cost[i][j] < min && cost[i][j] != INF)

                        min = cost[i][j];

                }

            }

        }

        for (i = 0; i < n; i++) {

            if (visited[i] && min == cost[i][j])

                a = i;

        }

        printf("%d-%d ", a, j);

        visited[j] = 1;

        ne++;

    }

}
```

```

        if (!visited[j] && cost[i][j] < min) {
            min = cost[i][j];
            a = u = i;
            b = v = j;
        }
    }
}

if (!visited[v]) {
    printf("Edge %d: (%d -> %d) Cost: %d\n", ne++, a, b, min);
    total += min;
    visited[b] = 1;
}
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 3 0
1 2 3 4
0 6 4 2
5 2 0 0

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

Minimum Cost = 7

==== Code Execution Successful ====

```