

```

#include<stdio.h>

#include<stdlib.h>

int i, j, k, a, b, u, v, n, ne = 1;

int min, mincost = 0, cost[9][9], parent[9];

int find(int);

int uni(int, int);

void main() {

    printf("\nImplementation of Kruskal's algorithm\n\n");

    printf("Enter the number of vertices: ");

    scanf("%d", &n);

    printf("\nEnter the cost adjacency matrix\n");

    for (i = 1; i <= n; i++) {

        for (j = 1; j <= n; j++) {

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

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

                cost[i][j] = 999;

        }

    }

    printf("\nThe edges of Minimum Cost Spanning Tree are\n\n");

    while (ne < n) {

        for (i = 1, min = 999; i <= n; i++) {

            for (j = 1; 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 (uni(u, v)) {
    printf("\n%d edge (%d,%d) = %d\n", ne++, a, b, min);
    mincost += min;
}
cost[a][b] = cost[b][a] = 999;
}

```

```

printf("\n\tMinimum cost = %d\n", mincost);
}

```

```

int find(int i) {
    while (parent[i])
        i = parent[i];
    return i;
}

```

```

int uni(int i, int j) {
    if (i != j) {
        parent[j] = i;
        return 1; // Return 1 if union operation is successful
    }
}

```

```
    return 0; // Return 0 if union operation is not performed  
}
```