26.Implementation of Minimum Spanning Tree using Kruskal Algorithm

```
#include <stdio.h>
#define MAX 30
int find(int i, int parent[]) {
   while (parent[i])
      i = parent[i];
   return i;
}
int uni(int i, int j, int parent[]) {
   if (i != j) {
      parent[j] = i;
      return 1;
   }
   return 0;
}
int main() {
   int n, i, j, u, v, a, b, min, mincost = 0, ne = 1;
   int cost[MAX][MAX];
   int parent[MAX] = {0};
   printf("Enter number of vertices: ");
   scanf("%d", &n);
   printf("Enter the adjacency matrix (use 0 if no edge):\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("\nEdges in the Minimum Spanning Tree:\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, parent);
        v = find(v, parent);
        if (uni(u, v, parent)) {
            printf("Edge %d: (%d - %d) cost: %d\n", ne++, a, b, min);
            mincost += min;
        }
        cost[a][b] = cost[b][a] = 999; // Remove edge
    }
    printf("\nMinimum Cost = %d\n", mincost);
    return 0;
}
```