

```
#include <stdio.h>
```

```
#define MAX 30
```

```
int parent[MAX];
```

```
// Function to find parent (with path compression)
```

```
int find(int i) {
```

```
    while (parent[i])
```

```
        i = parent[i];
```

```
    return i;
```

```
}
```

```
// Function to perform union
```

```
int unionNodes(int i, int j) {
```

```
    if (i != j) {
```

```
        parent[j] = i;
```

```
        return 1;
```

```
    }
```

```
    return 0;
```

```
}
```

```
int main() {
```

```
    int n, edges, u, v, a, b, i, j;
```

```
    int min, minCost = 0;
```

```
    int cost[MAX][MAX];
```

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

```
scanf("%d", &n);
```

```
printf("Enter the cost adjacency matrix (999 for no edge):\n");
```

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

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

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

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

```
    parent[i] = 0;
```

```
edges = 0;
```

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

```
while (edges < n - 1) {
```

```
    min = 999;
```

```
    for (i = 1; 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 (unionNodes(u, v)) {

    printf("Edge %d: (%d - %d) Cost: %d\n", edges + 1, a, b, min);

    minCost += min;

    edges++;

}

cost[a][b] = cost[b][a] = 999; // Mark as used

}

printf("\nMinimum Spanning Tree Cost = %d\n", minCost);

return 0;

}

```

The screenshot displays the Programiz Online C Compiler interface. The browser tabs at the top include '172.23.30.101', 'dharshini27/Data-Structure', 'Matrix multiplication program', and 'Online C Compiler - Programiz'. The address bar shows 'programiz.com/c-programming/online-compiler/'.

The main interface features a sidebar with icons for file management and a 'main.c' editor. The code in the editor is as follows:

```

49     b = v = j;
50     }
51     }
52 }
53
54 u = find(u);
55 v = find(v);
56
57 if (unionNodes(u, v)) {
58     printf("Edge %d: (%d - %d) Cost: %d\n", edges + 1, a, b,
59         min);
60     minCost += min;
61     edges++;
62 }
63
64 cost[a][b] = cost[b][a] = 999; // Mark as used
65 }
66
67 printf("\nMinimum Spanning Tree Cost = %d\n", minCost);
68 return 0;
69 }

```

Below the editor is a 'Run' button and a 'Share' link. To the right, the 'Output' panel shows the following text:

```

Enter number of vertices: 4
Enter the cost adjacency matrix (999 for no edge):
0 5 8 999
5 0 10 15
8 10 0 20
999 15 20 0

Edges in Minimum Spanning Tree:
Edge 1: (1 - 2) Cost: 5
Edge 2: (1 - 3) Cost: 8
Edge 3: (2 - 4) Cost: 15

Minimum Spanning Tree Cost = 28

=== Code Execution Successful ===

```