



```
1 #include<stdio.h>
2 void kruskals();
3 int cost[10][10],n,sum,min,i,j,count,k,u,v,parent[10],t[0][0];
4 void union_ij(int,int);
5 int find(int);
6 void main()
7 {
8     printf("Enter the Number of Vertices:");
9     scanf("%d", &n);
10    printf("Enter the Cost of Adjacency Matrix:\n");
11    for(i=0;i<n;i++)
12    {
13        for(j=0;j<n;j++)
14        {
15            scanf("%d", &cost[i][j]);
16        }
17    }
18    kruskals();
19 }
20 void kruskals()
21 {
22     count=0;
23     k=0;
24     sum=0;
25     for(i=0;i<n;i++)
26         parent[i]=i;
27     while(count!=n-1)
28     {
29         min=999;
30         for(i=0;i<n;i++)
31         {
32             for(j=0;j<n;j++)
33             {
34                 if(cost[i][j]<min && cost[i][j]!=0)
35                 {
36                     min=cost[i][j];
37                     u=i;
38                     v=j;
39                 }
40             }
41         }
42         i=find(u);
43         j=find(v);
44         if(i!=j)
45         {
46             t[k][0]=u;
47             t[k][1]=v;
48             k++;
49             count++;
50             sum=sum+cost[u][v];
51             union_ij(i,j);
52         }
53         cost[u][v]=cost[v][u]=999;
54     }
55     printf("Minimum Spanning Tree");
56     for(i=0;i<=n-1;i++)
57     {
58         printf("\n%d", t[i][0],t[i][1]);
59         printf("\nTotal Cost= %d", sum);
60     }
61 }
62 void union_ij(int i, int j)
63 {
64     if(i<j)
65         parent[j]=i;
66     else
67         parent[i]=j;
68 }
69 int find(int v)
70 {
71     while(parent[v]!=v)
72         v=parent[v];
73     return v;
74 }
```

× Terminal

```
Enter the Number of Vertices:4
Enter the Cost of Adjacency Matrix:
0 1 5 2
1 0 999 999
5 999 0 3
2 999 3 0
Minimum Spanning Tree
3-3
Total Cost= 6
```