

C 01knapsack.c C prims.c C kruskals.c X

```
C kruskals.c
1  /*To find Minimum Spanning tree using Kruskal's algorithm*/
2  #include <stdio.h>
3  #include<conio.h>
4  int cost[10][10],min,i,j,count,k,u,v,parent[10],t[10][10],n,sum=0;
5
6  void union_ij(int i,int j)
7  {
8      if (i<j)
9      {
10         parent[j]=i;
11     }
12     else
13     {
14         parent[i]=j;
15     }
16 }
17
18 int find(int v)
19 {
20     while (parent[v]!=v)
21     {
22         v=parent[v];
23     }
24     return v;
25 }
26
27 void krushkal()
28 {
29     count=0;
30     k=0;
31     sum=0;
32     for (i=0;i<n;i++)
33     {
34         parent[i]=i;
35     }
36     while (count!=(n-1))
37     {
38         min=999;
```

```
36 while (count!=(n-1))
37 {
38     min=999;
39     for (i=0;i<n;i++)
40     {
41         for (j=0;j<n;j++)
42         {
43             if (cost[i][j]<min && cost[i][j]!=0)
44             {
45                 min=cost[i][j];
46                 u=i;
47                 v=j;
48             }
49         }
50     }
51     i=find(u);
52     j=find(v);
53     if (i!=j)
54     {
55         t[k][0]=u;
56         t[k][1]=v;
57         k++;
58         count++;
59         sum=sum+cost[u][v];
60         union_ij(i,j);
61     }
62     cost[u][v]=cost[v][u]=999;
63 }
64 printf("Minimal spanning tree:\n");
65 for (i=0;i<k;i++)
66 {
67     printf("%d->%d ",t[i][0],t[i][1]);
68 }
69 printf("\nTotal cost=%d\n",sum);
70 }
71
72 int main()
73 {
```

C 01knapsack.c X C prims.c C kruskals.c X

C kruskals.c

```
58         count++;
59         sum=sum+cost[u][v];
60         union_ij(i,j);
61     }
62     cost[u][v]=cost[v][u]=999;
63 }
64 printf("Minimal spanning tree:\n");
65 for (i=0;i<k;i++)
66 {
67     printf("%d->%d  ",t[i][0],t[i][1]);
68 }
69 printf("\nTotal cost=%d\n",sum);
70 }
71
72 int main()
73 {
74     printf("Enter the number of vertices:\n");
75     scanf("%d",&n);
76     printf("Enter the cost adjacency matrix:\n");
77     for (i=0;i<n;i++)
78     {
79         for (j=0;j<n;j++)
80         {
81             scanf("%d",&cost[i][j]);
82         }
83     }
84     krushkal();
85     return 0;
86 }
```



C 01knapsack.c C prims.c C kruskals.c X

C kruskals.c

```
58     count++;  
59     sum=sum+cost[u][v];  
60     union_ij(i,j);  
61 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

powershell + v ^ X

```
PS C:\Users\muska\OneDrive\Desktop\C programs> gcc kruskals.c  
PS C:\Users\muska\OneDrive\Desktop\C programs> .\a.exe  
Enter the number of vertices:  
6  
Enter the cost adjacency matrix:  
0 3 999 999 6 5  
3 0 1 999 999 4  
999 1 0 6 999 4  
999 999 6 0 8 5  
6 999 999 8 0 2  
5 4 4 5 2 0  
Minimal spanning tree:  
1->2 4->5 0->1 1->5 3->5  
Total cost=15  
PS C:\Users\muska\OneDrive\Desktop\C programs> |
```