

prims-final.c++

```
1  #include <iostream>
2  using namespace std;
3
4  const int INFI = 99;
5  int adj_matrix[10][10];
6
7  void prims(int n) {
8      int mst[n-1][2], near[n+1], cost = 0;
9
10     for (int i=1; i<=n; i++)
11         near[i] = INFI;
12
13     int u,v,min = INFI;
14     for (int i=1; i<=n; i++) {
15         for (int j=i; j<=n; j++) {
16             if (i != j && adj_matrix[i][j] < min){
17                 min = adj_matrix[i][j];
18                 u = i;
19                 v = j;
20             }
21         }
22     }
23     cost += min;
24     mst[0][0] = u;
25     mst[0][1] = v;
26     near[u] = near[v] = 0;
27
28     for (int i=1; i<=n; i++) {
29         if (near[i] != 0) {
30             if (adj_matrix[i][u] < adj_matrix[i][v])
31                 near[i] = u;
32             else
33                 near[i] = v;
34         }
35     }
36
37     for (int i=1; i<n-1; i++) {
38         min = INFI;
39         int k;
40         for (int j=1; j<=n; j++) {
41             if (near[j] != 0 && adj_matrix[j][near[j]] < min) {
42                 min = adj_matrix[j][near[j]];
43                 k = j;
44             }
45         }
46         cost += min;
47         mst[i][0] = k;
48         mst[i][1] = near[k];
49         near[k] = 0;
50
51         for(int j=1; j<=n; j++) {
```

```
52         if (near[j] != 0 && adj_matrix[j][k] < adj_matrix[j][near[j]])
53             near[j] = k;
54     }
55 }
56
57
58 cout << "\nThe edges of minimum spanning tree:\n";
59 for (int i=0; i<n-1; i++)
60     cout << "(" << mst[i][0] << ", " << mst[i][1] << ")\n";
61 cout << "The cost of minimum spanning tree:\n" << cost << endl;
62 }
63
64 int main()
65 {
66     int n;
67     printf("\nEnter the number of vertices: ");
68     cin >> n;
69     printf("\nEnter the adjacency matrix: ");
70     for(int i=1; i<=n; i++)
71         for(int j=1; j<=n; j++)
72             cin >> adj_matrix[i][j];
73     prims(n);
74     return 0;
75 }
```