```cpp
#include<bits/stdc++.h>
using namespace std;

typedef pair<int, int> iPair;

struct Graph
{
    int V, E;
    vector< pair<int, iPair> > edges;

    Graph(int V, int E)
    {
        this->V = V;
        this->E = E;
    }

    void addEdge(int u, int v, int w)
    {
        edges.push_back({w, {u, v}});
    }

    int kruskalMST();
};

struct DisjointSets
{
    int *parent, *rnk;
    int n;
    DisjointSets(int n)
    {
        this->n = n;
        parent = new int[n+1];
        rnk = new int[n+1];
        for (int i = 0; i <= n; i++)
        {
            rnk[i] = 0;
            parent[i] = i;
        }
    }
    int find(int u)
    {
        if (u != parent[u])
            parent[u] = find(parent[u]);
        return parent[u];
    }
    void merge(int x, int y)
    {
        x = find(x), y = find(y);
        if (rnk[x] > rnk[y])
```

```
50              parent[y] = x;
51          else
52              parent[x] = y;
53
54          if (rnk[x] == rnk[y])
55              rnk[y]++;
56      }
57  };
58  int Graph::kruskalMST()
59  {
60      int mst_wt = 0;
61      sort(edges.begin(), edges.end());
62      DisjointSets ds(V);
63      vector< pair<int, iPair> >::iterator it;
64      for (it=edges.begin(); it!=edges.end(); it++)
65      {
66          int u = it->second.first;
67          int v = it->second.second;
68
69          int set_u = ds.find(u);
70          int set_v = ds.find(v);
71          if (set_u != set_v)
72          {
73              cout << u << " - " << v << endl;
74              mst_wt += it->first;
75              ds.merge(set_u, set_v);
76          }
77      }
78
79      return mst_wt;
80  }
81
82  int main()
83  {
84      int V = 9, E = 14;
85      Graph g(V, E);
86      g.addEdge(0, 1, 4);
87      g.addEdge(0, 7, 8);
88      g.addEdge(1, 2, 8);
89      g.addEdge(1, 7, 11);
90      g.addEdge(2, 3, 7);
91      g.addEdge(2, 8, 2);
92      g.addEdge(2, 5, 4);
93      g.addEdge(3, 4, 9);
94      g.addEdge(3, 5, 14);
95      g.addEdge(4, 5, 10);
96      g.addEdge(5, 6, 2);
97      g.addEdge(6, 7, 1);
98      g.addEdge(6, 8, 6);
```

```
 99      g.addEdge(7, 8, 7);
100
101      cout << "Edges of MST are \n";
102      int mst_wt = g.kruskalMST();
103
104      cout << "\nWeight of MST is " << mst_wt;
105
106      return 0;
107  }
108
```