Kruskal Algorithm

```
#include <iostream.h>
#include<conio.h>

int i,j,k,a,b,u,v,n,ne=1;
int min, mincost=0, cost[9][9], parent[9];
int find(int);
int uni(int,int);

void main()
{
   cout<<"\nImplementation of Kruskal's Algorithm\n";
   cout<<"\nEnter the number of vertices:";
   cin>>n;
   cout<<"\nEnter the cost adjacency matrix:\n";
   for(i=1;i<=n;i++)
   {
          for(j=1;j<=n;j++)
          {
             cin>>cost[i][j];
             if(cost[i][j]==0)
                    cost[i][j]=999;
          }
   }
   cout<<"The edges of Minimum Cost Spanning Tree are\n";
   while(ne < n)
   {
          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(uni(u,v))
          {
             cout<<ne++<<"edge =( "<<a<<","<<b<<")= "<<min<<endl;
             mincost +=min;
          }
          cost[a][b]=cost[b][a]=999;
   }
   cout<<"\n\tMinimum cost = \n"<<mincost;
```

```
    getch();
}

int find(int i)
{
    while(parent[i])
              i=parent[i];
    return i;
}

int uni(int i,int j)
{
    if(i!=j)
    {
        parent[j]=i;
        return 1;
    }
    return 0;
}
```