Kruskals

```
#include <stdio.h>
#define I 65535

int edge[9][3]={{1,2,28},{1,6,10},{2,3,16},{2,7,14},{3,4,12},
{4,5,22},{4,7,18},{5,6,25},{5,7,24}};
int set[8]={-1,-1,-1,-1,-1,-1,-1,-1};
int included[9]={0,0,0,0,0,0,0,0,0};
void join(int u,int v)
{
    if(set[u]<set[v])
    {
        set[u]+=set[v];

        set[v]=u;
    }
    else
    {
        set[v]+=set[u];
        set[u]=v;

    }
}
int find(int u)
{
    int x=u,v=0;
    while(set[x]>0)
    {
        x=set[x];
    }
    while(u!=x)
    {
        v=set[u];
        set[u]=x;
        u=v;
    }

    return x;
}
int t[2][7];
int main(int argc, const char * argv[])
{
    int u=0,v=0,i,j,k=0,min=65535,n=9;
    i=0;

    while(i<6)
    {
        min=65535;
        for(j=0;j<n;j++)
        {
            if(included[j]==0 && edge[j][2]<min)
            {
                u=edge[j][0];v=edge[j][1];min=edge[j][2];
                k=j;
            }
        }
        if(find(u)!= find(v))
        {
            t[0][i]=u;t[1][i]=v;
            join(find(u),find(v));
            included[k]=1;
            i++;
           // printf("%d  %d %d %d\n",u,v,find(u),find(v));

        }
        else
        {
            included[k]=1;
        }
    }
    printf("Spanning Tree\n");
    for(i=0;i<6;i++)
    {
        printf("%d %d\n",t[0][i],t[1][i]);
    }
    return 0;
}
```