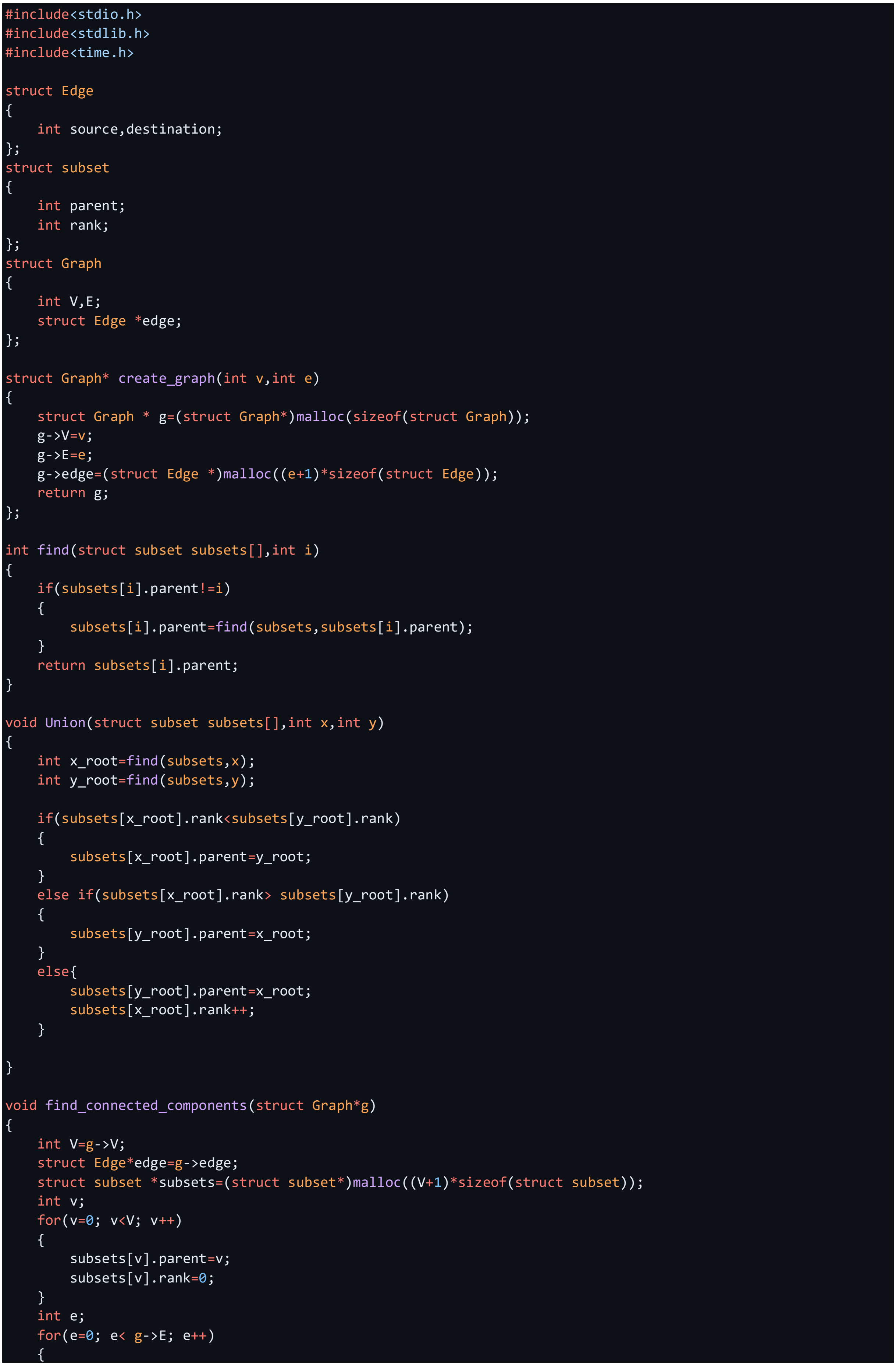
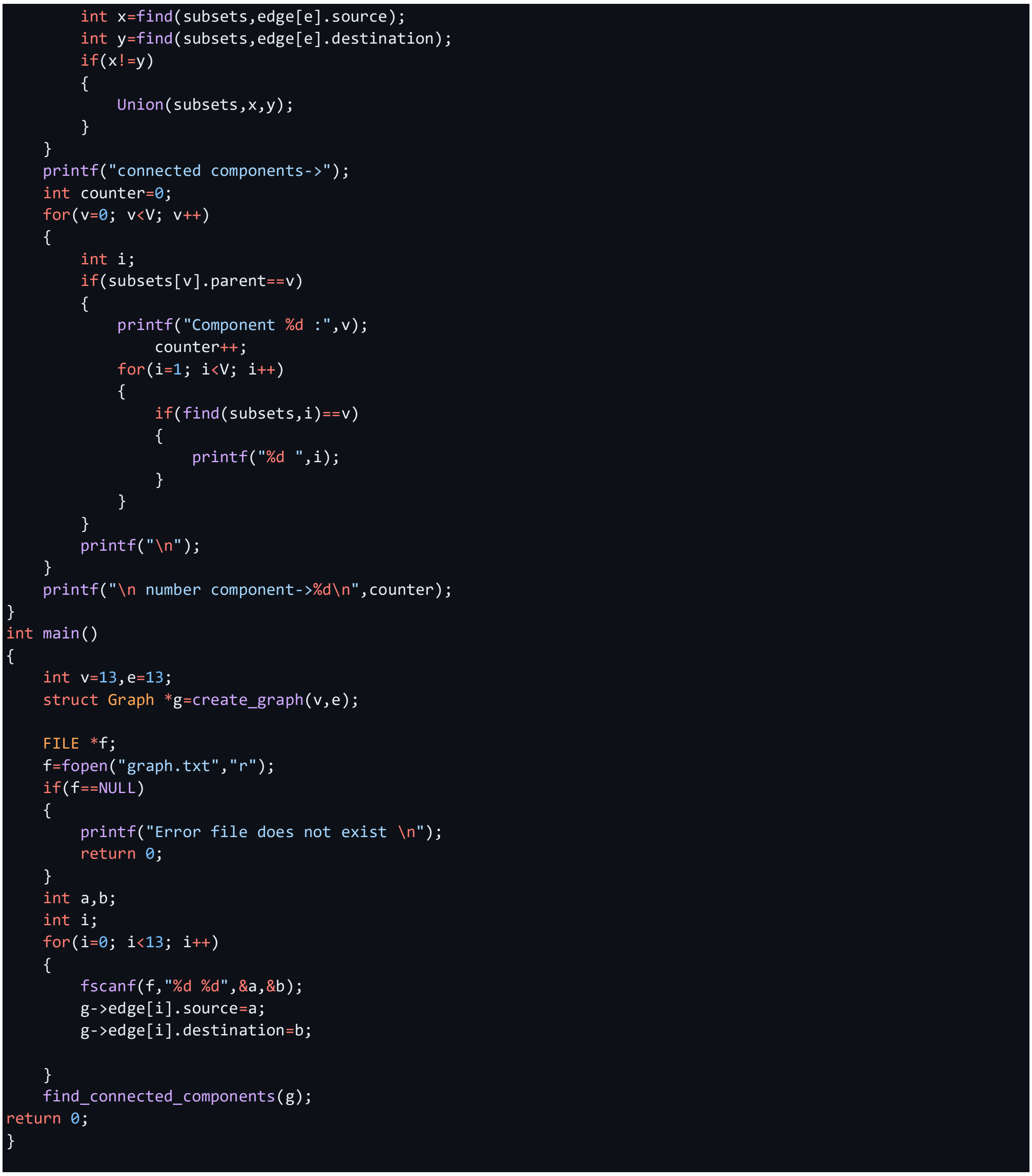


```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>

struct Edge
{
    int source,destination;
};
struct subset
{
    int parent;
    int rank;
};
struct Graph
{
    int V,E;
    struct Edge *edge;
};

struct Graph* create_graph(int v,int e)
{
    struct Graph * g=(struct Graph*)malloc(sizeof(struct Graph));
    g->V=v;
    g->E=e;
    g->edge=(struct Edge *)malloc((e+1)*sizeof(struct Edge));
    return g;
};

int find(struct subset subsets[],int i)
{
    if(subsets[i].parent!=i)
    {
        subsets[i].parent=find(subsets,subsets[i].parent);
    }
    return subsets[i].parent;
}

void Union(struct subset subsets[],int x,int y)
{
    int x_root=find(subsets,x);
    int y_root=find(subsets,y);

    if(subsets[x_root].rank<subsets[y_root].rank)
    {
        subsets[x_root].parent=y_root;
    }
    else if(subsets[x_root].rank> subsets[y_root].rank)
    {
        subsets[y_root].parent=x_root;
    }
    else{
        subsets[y_root].parent=x_root;
        subsets[x_root].rank++;
    }

}

void find_connected_components(struct Graph*g)
{
    int V=g->V;
    struct Edge*edge=g->edge;
    struct subset *subsets=(struct subset*)malloc((V+1)*sizeof(struct subset));
    int v;
    for(v=0; v<V; v++)
    {
        subsets[v].parent=v;
        subsets[v].rank=0;
    }
    int e;
    for(e=0; e< g->E; e++)
    {
```



```c
        int x=find(subsets,edge[e].source);
        int y=find(subsets,edge[e].destination);
        if(x!=y)
        {
            Union(subsets,x,y);
        }
    }
    printf("connected components->");
    int counter=0;
    for(v=0; v<V; v++)
    {
        int i;
        if(subsets[v].parent==v)
        {
            printf("Component %d :",v);
                counter++;
            for(i=1; i<V; i++)
            {
                if(find(subsets,i)==v)
                {
                    printf("%d ",i);
                }
            }
        }
        printf("\n");
    }
    printf("\n number component->%d\n",counter);
}
int main()
{
    int v=13,e=13;
    struct Graph *g=create_graph(v,e);

    FILE *f;
    f=fopen("graph.txt","r");
    if(f==NULL)
    {
        printf("Error file does not exist \n");
        return 0;
    }
    int a,b;
    int i;
    for(i=0; i<13; i++)
    {
        fscanf(f,"%d %d",&a,&b);
        g->edge[i].source=a;
        g->edge[i].destination=b;

    }
    find_connected_components(g);
return 0;
}
```