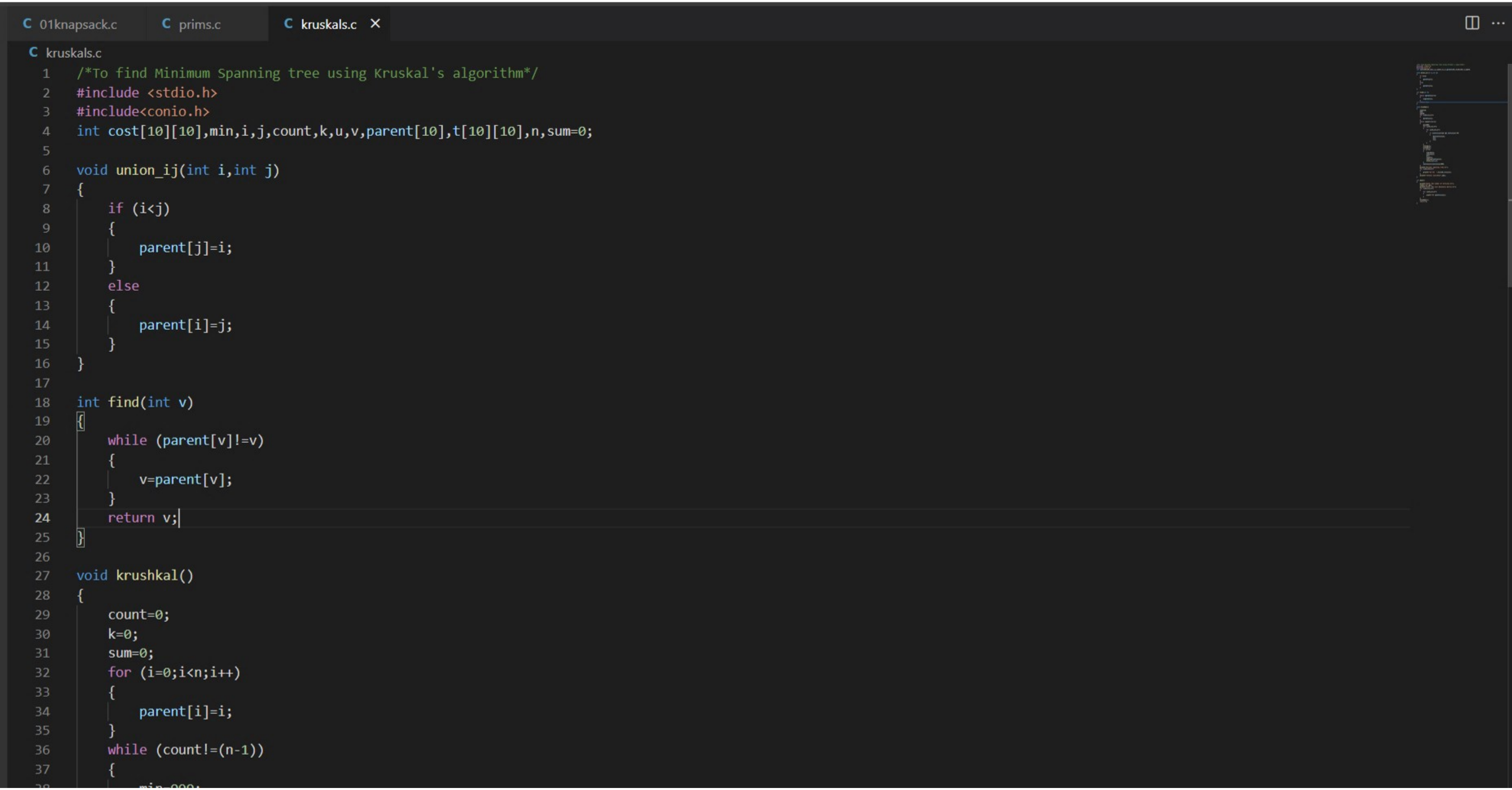
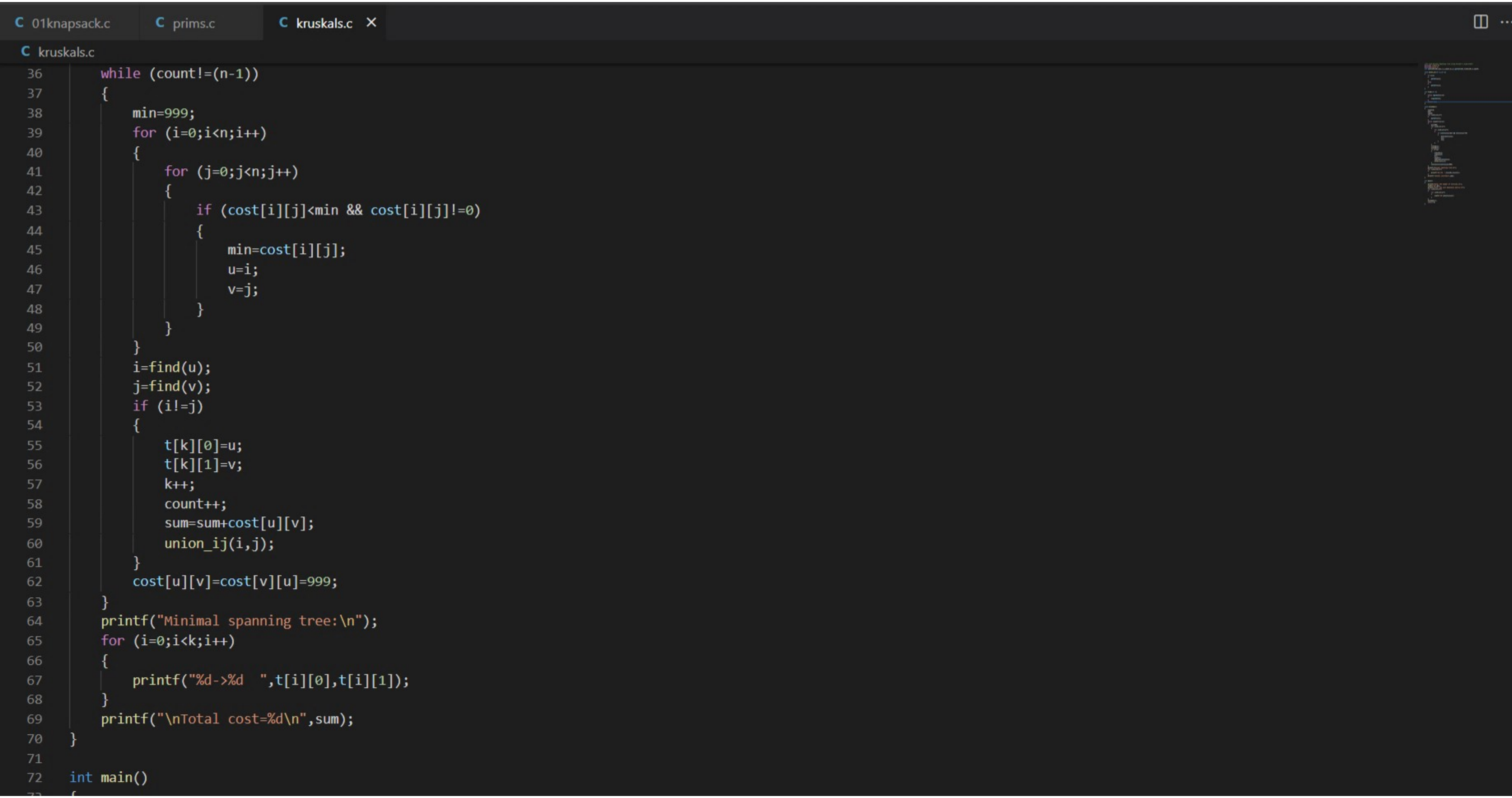
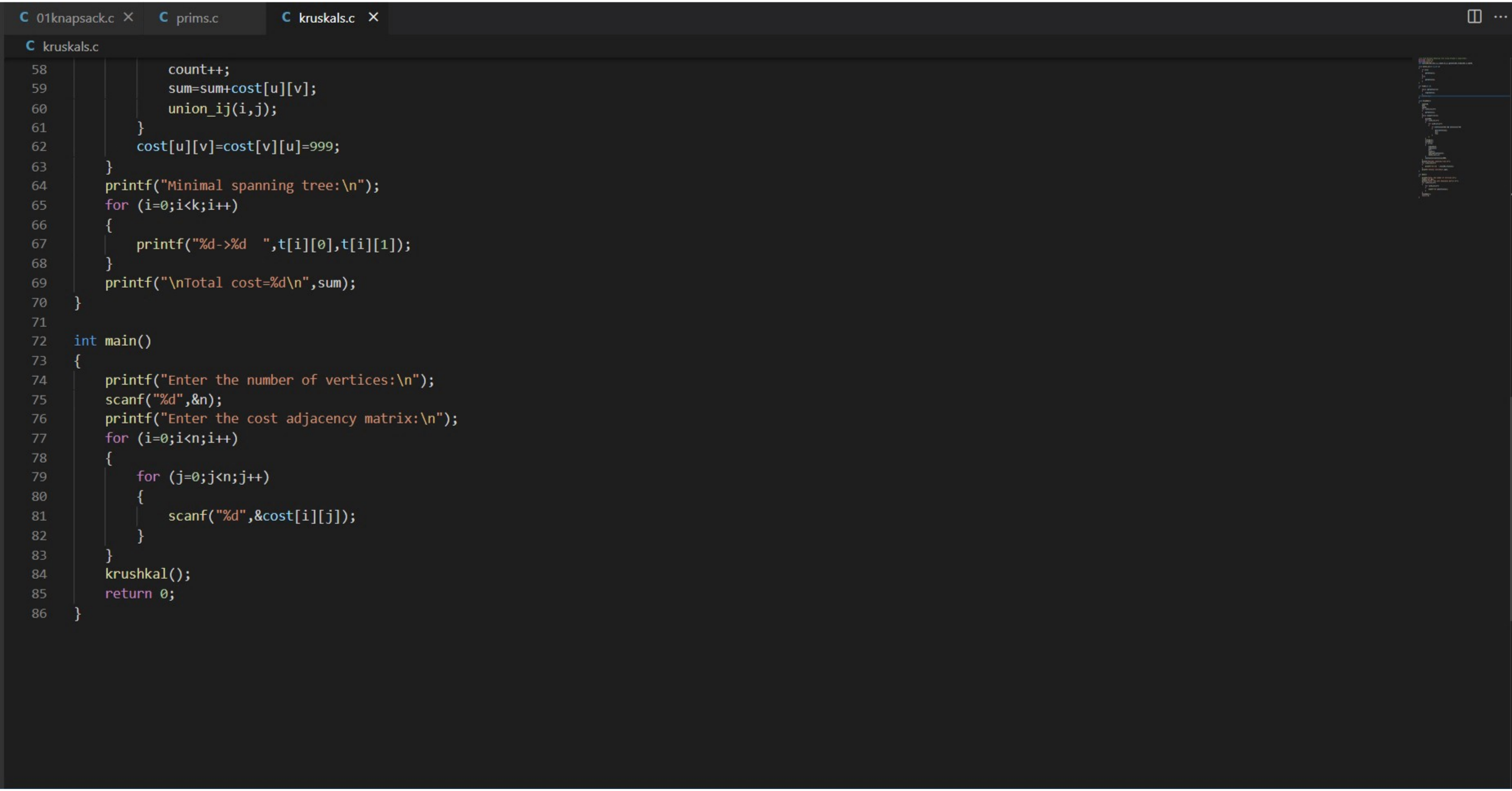


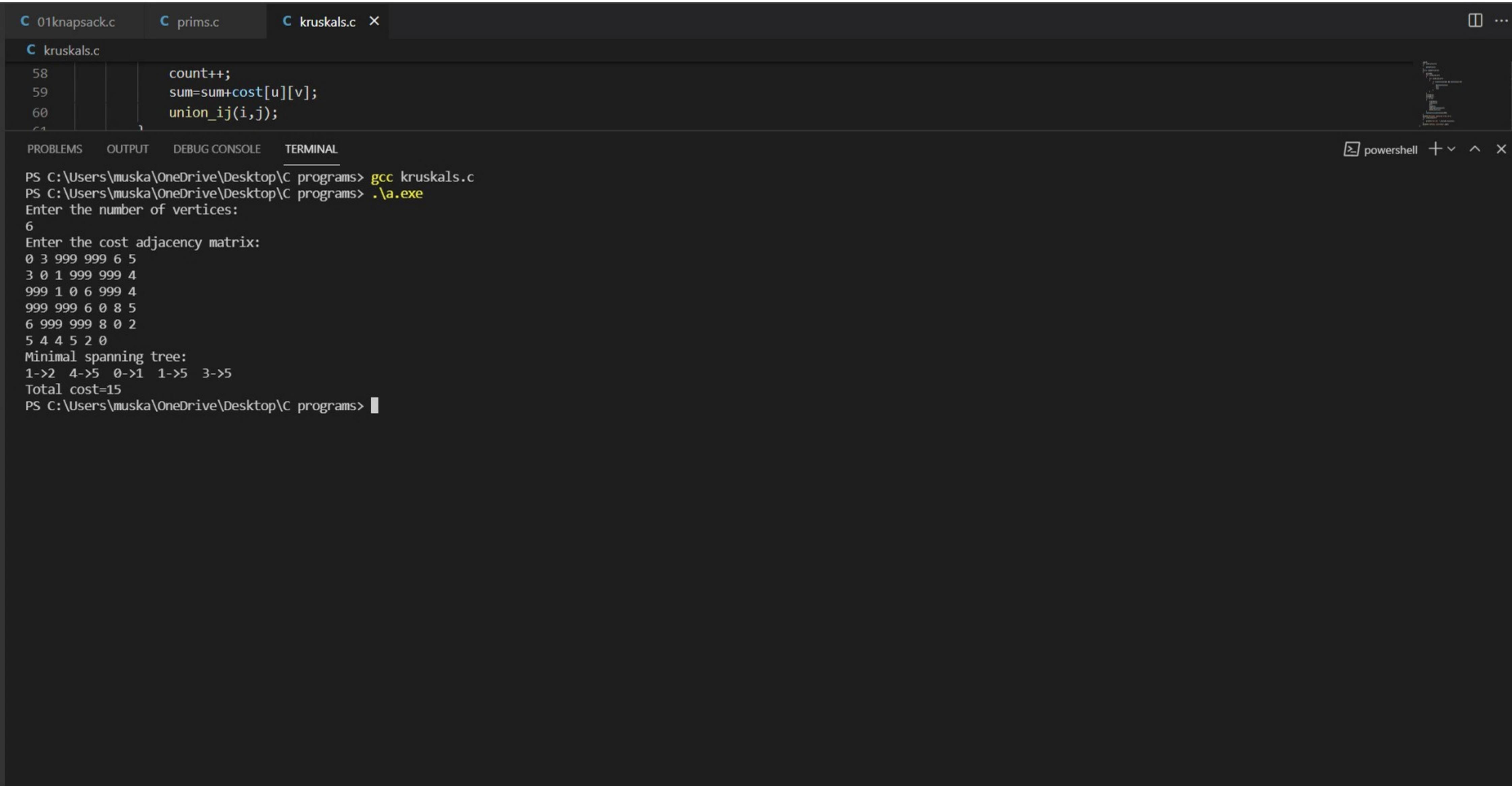
```c
/*To find Minimum Spanning tree using Kruskal's algorithm*/
#include <stdio.h>
#include<conio.h>
int cost[10][10],min,i,j,count,k,u,v,parent[10],t[10][10],n,sum=0;

void union_ij(int i,int j)
{
    if (i<j)
    {
        parent[j]=i;
    }
    else
    {
        parent[i]=j;
    }
}

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

void krushkal()
{
    count=0;
    k=0;
    sum=0;
    for (i=0;i<n;i++)
    {
        parent[i]=i;
    }
    while (count!=(n-1))
    {
        min=999;
```



```c
    while (count!=(n-1))
    {
        min=999;
        for (i=0;i<n;i++)
        {
            for (j=0;j<n;j++)
            {
                if (cost[i][j]<min && cost[i][j]!=0)
                {
                    min=cost[i][j];
                    u=i;
                    v=j;
                }
            }
        }
        i=find(u);
        j=find(v);
        if (i!=j)
        {
            t[k][0]=u;
            t[k][1]=v;
            k++;
            count++;
            sum=sum+cost[u][v];
            union_ij(i,j);
        }
        cost[u][v]=cost[v][u]=999;
    }
    printf("Minimal spanning tree:\n");
    for (i=0;i<k;i++)
    {
        printf("%d->%d  ",t[i][0],t[i][1]);
    }
    printf("\nTotal cost=%d\n",sum);
}

int main()
{
```



```c
            count++;
            sum=sum+cost[u][v];
            union_ij(i,j);
        }
        cost[u][v]=cost[v][u]=999;
    }
    printf("Minimal spanning tree:\n");
    for (i=0;i<k;i++)
    {
        printf("%d->%d  ",t[i][0],t[i][1]);
    }
    printf("\nTotal cost=%d\n",sum);
}

int main()
{
    printf("Enter the number of vertices:\n");
    scanf("%d",&n);
    printf("Enter the cost adjacency matrix:\n");
    for (i=0;i<n;i++)
    {
        for (j=0;j<n;j++)
        {
            scanf("%d",&cost[i][j]);
        }
    }
    krushkal();
    return 0;
}
```



```
            count++;
            sum=sum+cost[u][v];
            union_ij(i,j);
        }
```

```
PS C:\Users\muska\OneDrive\Desktop\C programs> gcc kruskals.c
PS C:\Users\muska\OneDrive\Desktop\C programs> .\a.exe
Enter the number of vertices:
6
Enter the cost adjacency matrix:
0 3 999 999 6 5
3 0 1 999 999 4
999 1 0 6 999 4
999 999 6 0 8 5
6 999 999 8 0 2
5 4 4 5 2 0
Minimal spanning tree:
1->2  4->5  0->1  1->5  3->5
Total cost=15
PS C:\Users\muska\OneDrive\Desktop\C programs> 
```