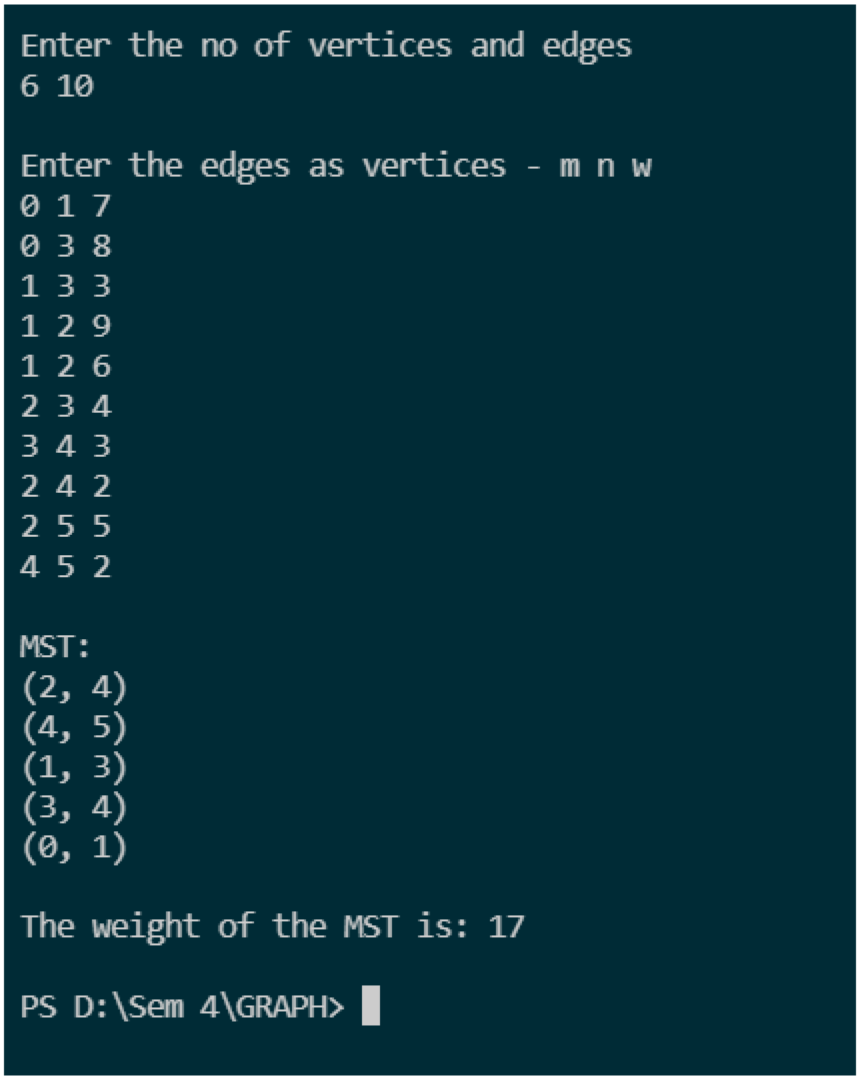
Program statement: Write programs in C to find out the minimum spanning tree of a given undirected graph using the following: 1) Kruskal's Algorithm

Source code:

```c
#include<stdio.h>
#include<string.h>

int v, e, parent[20];

typedef struct edge{
    int u, v, w;
}edge;

void sort(edge edges[e])
{
    int i, j;
    edge key;
    for (i = 1; i < e; i++) {
        key = edges[i];
        j = i - 1;
        while (j >= 0 && edges[j].w > key.w){
            edges[j + 1] = edges[j];
            j = j - 1;
        }
        edges[j + 1] = key;
    }
}

int find(int n)
{
    while( parent[n]!=-1 )
    {
        n = parent[n];
    }
    return n;
}

void union_(int m, int n)
{
    int mp,np;
    mp=find(m);
    np=find(n);
    parent[np]=mp;
}

void kruskal(edge edges[e])
{
    edge mst[v-1];
    int mstl=-1, i, mstw=0;
    sort(edges);
    for( i=0; i<e; i++ )
    {
        if(find(edges[i].u) != find(edges[i].v))
        {
            mst[++mstl] = edges[i];
            union_(find(edges[i].u), find(edges[i].v));
        }
    }
    printf("\nMST:\n");
    for( i=0; i<=mstl; i++ )
    {
        printf("(%d, %d)\n", mst[i].u, mst[i].v);
        mstw += mst[i].w;
    }
    printf("\nThe weight of the MST is: %d\n ", mstw);
}

int main()
{
    int i, m, n, w;
    printf("Enter the no of vertices and edges\n");
    scanf("%d%d", &v, &e);
    edge edges[e];
    for( i=0; i<v; i++ )
    {
        parent[i]=-1;
    }
    printf("\nEnter the edges as vertices - m n w\n");
    for( i=0; i<e; i++ )
    {
        scanf("%d%d%d", &edges[i].u, &edges[i].v, &edges[i].w);
    }
    kruskal(edges);
}
```

Output:



```
Enter the no of vertices and edges
6 10

Enter the edges as vertices - m n w
0 1 7
0 3 8
1 3 3
1 2 9
1 2 6
2 3 4
3 4 3
2 4 2
2 5 5
4 5 2

MST:
(2, 4)
(4, 5)
(1, 3)
(3, 4)
(0, 1)

The weight of the MST is: 17

PS D:\Sem 4\GRAPH>
```

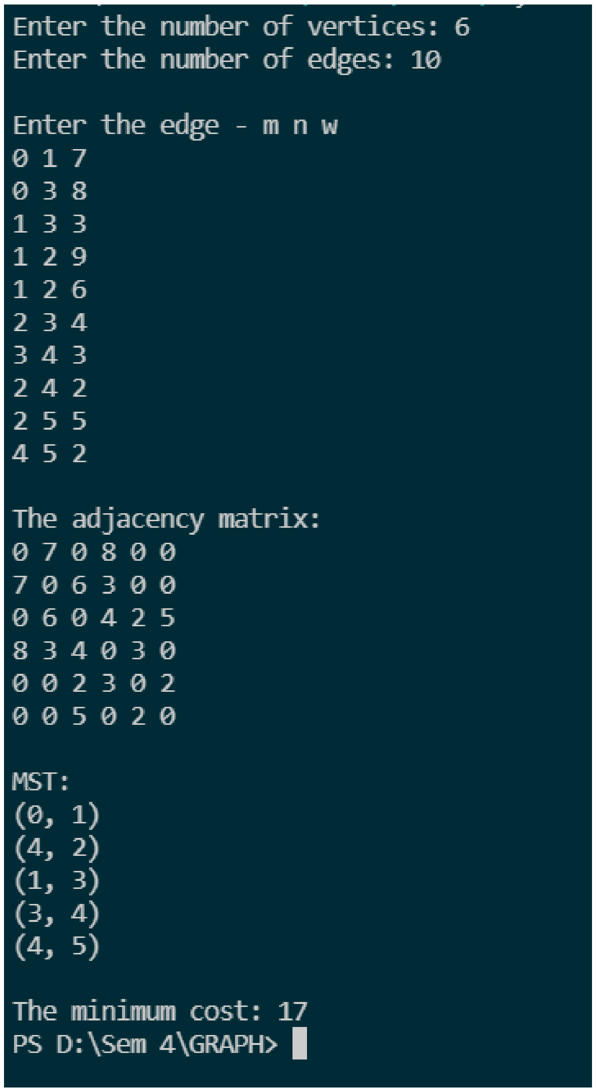
Program statement: 2) Prim's Algorithm.

Source code:

```
// A C Program for prim algorithm
// representation of graphs using adjacency matrix
#include <stdio.h>
```

```c
#include <stdlib.h>

void prim(int v, int a[v][v])
{
    int i,j,min,n,sw=0;
    int parent[v], key[v], mstset[v];
    for( j=0; j<v; j++ )
    {
        key[j]=9999;
        mstset[j]=0;
        parent[j]=-1;
    }
    j=0;
    key[j]=0;
    while (j<v)
    {
        min=9998;
        for( i=0; i<v; i++ )
        {
            if( mstset[i]==0 )
            {
                if( min>key[i] )
                {
                    min=key[i];
                    n=i;
                }
            }
        }
        mstset[n]=1;
        for( i=0; i<v; i++ )
        {
            if( a[n][i] > 0 )
            {
                if( mstset[i]==0 )
                {
                    if( key[i]>a[n][i] )
                    {
                        key[i]=a[n][i];
                        parent[i]=n;
                    }
                    }
                }
            }
            j++;
        }
    printf("\nMST:\n");
    for( i=1; i<v; i++ )
    {
        sw += a[i][parent[i]];
        printf("(%d, %d)\n",parent[i],i);
    }
    printf("\nThe minimum cost: %d\n",sw);
}

int main()
{
    int v,e;
    printf("Enter the number of vertices: ");
    scanf("%d",&v);
    printf("Enter the number of edges: ");
    scanf("%d",&e);
    int a[v][v];
    int i,j,p,w;
    for( i=0; i<v; i++ )
    {
        for( j=0; j<v; j++ )
        {
            a[i][j]=0;
        }
    }
    printf("\nEnter the edge - m n w\n");
    for( p=0; p<e; p++ )
    {
        scanf("%d%d%d", &i, &j, &w);
        if(( a[i][j] > 0 && a[i][j] < w) || ( i==j ))
            continue;
        else{
            a[i][j]=w;
            a[j][i]=w;
        }
    }
    printf("\nThe adjacency matrix:\n");
    for( i=0; i<v; i++)
    {
        for( j=0; j<v; j++)
        {
            printf("%d ",a[i][j]);
        }
        printf("\n");
    }
    prim(v,a);
}
```

Output:



```
Enter the number of vertices: 6
Enter the number of edges: 10

Enter the edge - m n w
0 1 7
0 3 8
1 3 3
1 2 9
1 2 6
2 3 4
3 4 3
2 4 2
2 5 5
4 5 2

The adjacency matrix:
0 7 0 8 0 0
7 0 6 3 0 0
0 6 0 4 2 5
8 3 4 0 3 0
0 0 2 3 0 2
0 0 5 0 2 0

MST:
(0, 1)
(4, 2)
(1, 3)
(3, 4)
(4, 5)

The minimum cost: 17
PS D:\Sem 4\GRAPH>
```