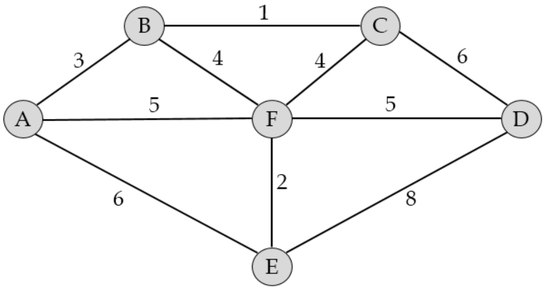
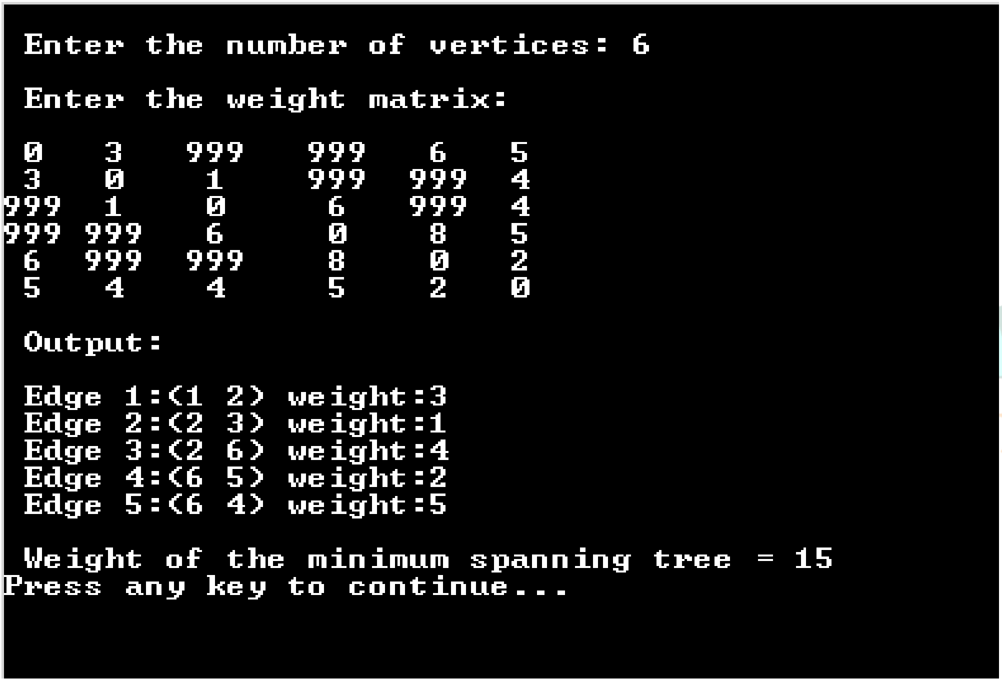
```
/*C Program to implement Prim’s Algorithm
Input  :  1. No. of vertices in the graph
             2. Weight Matrix
Output : 1. Edges with the minimum weight associated with them
              2. Weight of the minimum spanning tree
*/
#include<stdio.h>

 int a,b,u,v,n,i,j,ne=1;
 int visited[10]={0},min,minWeight=0,weight[10][10];

 int main()
  {
        printf("\n Enter the number of vertices: ");
        scanf("%d",&n);
        printf("\n Enter the weight matrix:\n\n");
        for(i=1;i<=n;i++)
           for(j=1;j<=n;j++)
                {
                        scanf("%d",&weight[i][j]);
                        if(weight[i][j]==0)
                           weight[i][j]=999;
                }
                visited[1]=1;
                printf("\n");

   printf(" Output:\n");

  while(ne < n)
        {
                for(i=1,min=999;i<=n;i++)
```

```
    for(j=1;j<=n;j++)
        if(weight[i][j]< min)
        if(visited[i]!=0)
        {
            min=weight[i][j];
          a=u=i;
            b=v=j;
        }

        if(visited[u]==0 || visited[v]==0)
        {
          printf("\n Edge %d:(%d %d) weight:%d",ne++,a,b,min);
          minWeight+=min;
          visited[b]=1;
    }
        weight[a][b]=weight[b][a]=999;
}
printf("\n");
printf("\n Weight of the minimum spanning tree = %d",minWeight);
printf("\n");
}
```

Sample Input and Output:





```
 Enter the number of vertices: 6

 Enter the weight 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

 Output:

 Edge 1:(1 2) weight:3
 Edge 2:(2 3) weight:1
 Edge 3:(2 6) weight:4
 Edge 4:(6 5) weight:2
 Edge 5:(6 4) weight:5

 Weight of the minimum spanning tree = 15
Press any key to continue...
```