```c
//Prims
#include<stdio.h>
#include<stdlib.h>
int a,b,u,t,m,n,i,j;
int v[10],c=1,min=0,cost[10][10];
int main(){
    printf("Enter the number of vertices :- ");
    scanf("%d",&n);
    printf("\nEnter the adjacency matrix:\n");
    for (i=1;i<=n;i++){
        for (j=1;j<=n;j++) {
            scanf("%d",&cost[i][j]);
            if(cost[i][j]==0)
                cost[i][j]=999;
        }
    }
    v[1]=1;
    while(c<n){
        for (i=1,m=999;i<=n;i++){
            for (j=1;j<=n;j++){
                if(cost[i][j]<m){
                    if(v[i]!=0){
                        m=cost[i][j];
                        a=u=i;
                        b=t=j;
                    }
```

```
1139                        b=t=j;
1140                    }
1141                }
1142            }
1143        }
1144        if(v[u]==0 || v[t]==0){
1145            printf("\nEdge %d:(%d %d) cost:%d",c++,a,b,m);
1146            min+=m;
1147            v[b]=1;
1148        }
1149        cost[a][b]=cost[b][a]=999;
1150    }
1151    printf("\nMinimun cost=%d\n",min);
1152    return 0;
1153 }
1154
1155
```

```
Enter the number of vertices :- 3

Enter the adjacency matrix:
0 2 5
2 0 0
5 0 0

Edge 1:(1 2) cost:2
Edge 2:(1 3) cost:5
Minimun cost=7

Process returned 0 (0x0)   execution time : 21.159 s
Press any key to continue.
```

```c
//Kruskal
#include<stdio.h>
#include<stdlib.h>
int i,j,k,a,b,u,v,n,ne=1;
int min,mincost=0,cost[10][10],parent[10];
int main()
{
    printf("Enter the no. of vertices :- ");
    scanf("%d",&n);
    printf("\nEnter the adjacency matrix :\n");
    for(i=1;i<=n;i++){
        for(j=1;j<=n;j++){
            scanf("%d",&cost[i][j]);
            if(cost[i][j]==0)
                cost[i][j]=999;
        }
    }
    while(ne < n){
        for(i=1,min=999;i<=n;i++){
            for(j=1;j <= n;j++){
                if(cost[i][j] < min){
                    min=cost[i][j];
                    a=u=i;
                    b=v=j;
                }
            }
        }
        while(parent[u])
            u=parent[u];
```

```
1185            while(parent[u])
1186                u=parent[u];
1187            while(parent[v])
1188                v=parent[v];
1189            if(u!=v){
1190                printf("Edge %d:%d,%d) Cost:%d\n",ne++,a,b,min);
1191                mincost+=min;
1192                parent[v]=u;
1193            }
1194            cost[a][b]=cost[b][a]=999;
1195        }
1196        printf("\nMinimum cost = %d\n",mincost);
1197        return 0;
1198    }
1199
1200
1201
1202
1203
```

```
Enter the no. of vertices :- 6

Enter the adjacency matrix :
0 3 1 6 0 0
3 0 5 0 3 0
1 5 0 5 6 4
6 0 5 0 0 2
0 3 6 0 0 6
0 0 4 2 6 0
Edge 1:(1,3) Cost:1
Edge 2:(4,6) Cost:2
Edge 3:(1,2) Cost:3
Edge 4:(2,5) Cost:3
Edge 5:(3,6) Cost:4

Minimum cost = 13

Process returned 0 (0x0)   execution time : 48.654 s
Press any key to continue.
```