

CODE for PRIMS Algorithm: -

```
#include<stdio.h>
#include<conio.h>
int a,b,u,v,n,i,j,ne=1;
int visited[10]= {0},min,mincost=0,cost[10][10];
void main()
{
    printf("\n Enter the number of nodes:");
    scanf("%d",&n);
    printf("\n Enter 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;
        }
    visited[1]=1;

    printf("\n");
    while(ne<n)
    {
        for (i=1,min=999;i<=n;i++)
            for (j=1;j<=n;j++)
                if(cost[i][j]<min)
                    if(visited[i]!=0)
                    {
                        min=cost[i][j];
                        a=u=i;
                        b=v=j;
                    }
        if(visited[u]==0 || visited[v]==0)
        {
            printf("\n Edge %d:(%d %d) cost:%d",ne++,a,b,min);
            mincost+=min;
            visited[b]=1;
        }
        cost[a][b]=cost[b][a]=999;
    }
    printf("\n Minimun cost=%d",mincost);
    getch();
}
```



```
C:\Users\Mandavi Dubey\Desktop>gcc exp1.c
```

```
C:\Users\Mandavi Dubey\Desktop>a
```

Enter the number of nodes:7

Enter the adjacency matrix:

```
0 28 0 0 0 10 0
28 0 16 0 0 0 14
0 16 0 12 0 0 0
0 0 12 0 22 0 18
0 0 0 22 0 25 24
10 0 0 0 25 0 0
0 14 0 18 24 0 0
```

Edge 1:(1 6) cost:10

Edge 2:(6 5) cost:25

Edge 3:(5 4) cost:22

Edge 4:(4 3) cost:12

Edge 5:(3 2) cost:16

Edge 6:(2 7) cost:14

Minimun cost=99

CODE for Kruskal's Algorithm: -

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
int i,j,k,a,b,u,v,n,ne=1;
int min,mincost=0,cost[9][9],parent[9];
int find(int);
int uni(int,int);
void main()
{
printf("Implementation of Kruskal's algorithm \n");
printf("Enter the no. of vertices: \n");
scanf("%d",&n);
printf("Enter the cost 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;
}
}
printf("The edges of Minimum Cost Spanning Tree are: \n");
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;
}
}
}
u=find(u);
v=find(v);
if(uni(u,v))
{
printf("%d edge (%d,%d) =%d \n",ne++,a,b,min);
mincost +=min;
}
cost[a][b]=cost[b][a]=999;
}
printf("\nMinimum cost = %d \n",mincost);
getch();
}
int find(int i)
```



```

{
    while (parent[i])
        i=parent[i];
    return i;
}
int uni(int i,int j)
{
    if(i!=j)
    {
        parent[j]=i;
        return 1;
    }
    return 0;
}

```

Output obtained from the above code for Kruskal's Algorithm

C:\Users\Mandavi Dubey\Desktop>gcc exp1.c

C:\Users\Mandavi Dubey\Desktop>a

Implementation of Kruskal's algorithm

Enter the no. of vertices:

7

Enter the cost adjacency matrix:

0 28 0 0 0 10 0

28 0 16 0 0 0 14

0 16 0 12 0 0 0

0 0 12 0 22 0 18

0 0 0 22 0 25 24

10 0 0 0 25 0 0

0 14 0 18 24 0 0

The edges of Minimum Cost Spanning Tree are:

1 edge (1,6) =10

2 edge (3,4) =12

3 edge (2,7) =14

4 edge (2,3) =16

5 edge (4,5) =22

6 edge (5,6) =25

tMinimum cost = 99