KRUSHKAL PROGRAM:-

PROGRAM :-

```
#include<stdio.h>
void kruskals();
int cost[10][10],n,min,i,j,sum,count,k,u,v,parent[10];
int t[10][10];
void union_ij(int,int);
int find(int);
void main()
{
printf("Enter the number of vertices\n");
scanf("%d",&n);
printf("Enter the cost of adjacency matrix\n");
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
scanf("%d", &cost[i][j]);
}
}
kruskals();
}

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

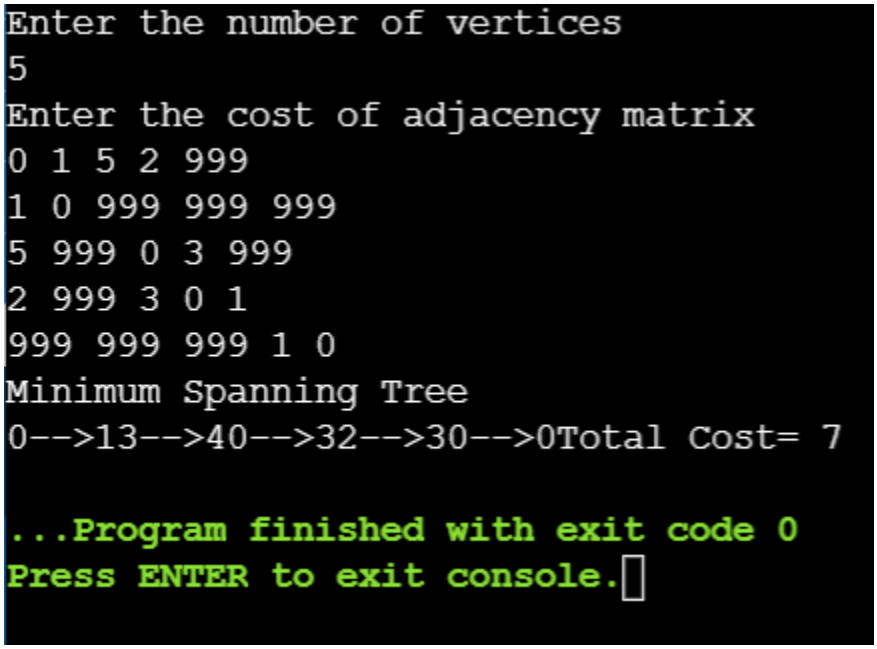
```
{
min=999;
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
if(cost[i][j]<min && cost[i][j]!=0)
{
min=cost[i][j];
u=i;
v=j;
}
}
}
i=find(u);
j=find(v);
if(i!=j)
{
t[k][0]=u;
t[k][1]=v;
k++;
count++;
sum=sum+cost[u][v];
union_ij(i,j);
}
cost[u][v]=cost[v][u]=999;
}
printf("Minimum Spanning Tree\n");
for(i=0;i<=n-1;i++)
{
```

```
}
printf("Minimum Spanning Tree\n");
for(i=0;i<=n-1;i++)
{
printf("%d-->%d",t[i][0],t[i][1]);
}
printf("Total Cost= %d", sum);
}

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;
}
```

OUTPUT:-


Enter the number of vertices
5
Enter the cost of adjacency matrix
0 1 5 2 999
1 0 999 999 999
5 999 0 3 999
2 999 3 0 1
999 999 999 1 0
Minimum Spanning Tree
0-->13-->40-->32-->30-->0Total Cost= 7
...Program finished with exit code 0
Press ENTER to exit console.