**Program 9:** **Write and execute a program to find Minimum Spanning Tree using Kruskal’s method**

include<stdio.h>

#include<conio.h>

int parent[20]={0},min,mincost=0,ne=1,n,cost[20][20];

int a,b,i,j,u,v;

void kruskal(void);

void main()

{

clrscr();

printf("Enter the number of nodes:");

scanf("%d",&n);

printf("Enter the cost matrix:");

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;

}

kruskal();

getch();

}

void kruskal()

{

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];

while(parent[v])

v=parent[v];

if(u!=v)

{

printf("%d\tedge\t(%d,%d)=%d\n",ne++,a,b,min);

mincost+=min;

parent[v]=u;

}

cost[a][b]=cost[b][a]=999;

}

printf("The minimum cost=%d\n",mincost);

}

/\*

Enter the number of nodes:5

Enter the cost matrix:

OR

#include<stdio.h>

int parent[20] = {0}, min, mincost = 0, ne = 1, n, cost[20][20], a, b, i, j, u, v;

// Kruskal Function

void kruskal() {

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];

while(parent[v])

v = parent[v];

if(u != v) {

printf("%d\tedge\t(%d,%d) = %d\n", ne++, a, b, min);

mincost += min;

parent[v] = u;

}

cost[a][b] = cost[b][a] = 999;

}

printf("The minimum cost = %d\n",mincost);

}

// Main Program

int main() {

printf("Enter the number of nodes:\n");

scanf("%d",&n);

printf("Enter the cost 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;

}

}

kruskal();

}

0 3 2 9 999

3 0 4 5 999

2 4 0 6 7

9 5 6 0 8

999 999 7 8 0

1 edge (1,3)=2

2 edge (1,2)=3

3 edge (2,4)=5

4 edge (3,5)=7

The minimum cost=17

\*/