**PROGRAM:**

#include <stdio.h>

void swapint(int\*, int\*);

void makeset(int), unionp(int, int);

int findset(int);

void link(int, int);

int parent[100], rank[100] = {};

int main() {

int mst[100][100] = {};

int adj[100][100];

int edges[100][3];

int v = 0, e = 0;

printf("Enter number of vertices : ");

scanf("%d", &v);

printf("Enter the adjacency matrix : \n");

for(int i = 0; i < v; i++) {

for(int j = 0; j < v; j++) {

scanf("%d", &adj[i][j]);

}

}

printf("The graph is : \n");

for(int i = 0; i < v; i++) {

for(int j = 0; j < v; j++) {

printf("%d ", adj[i][j]);

}

printf("\n");

}

printf("\nThe edges are : \n");

for(int i = 0; i < v; i++) {

for(int j = i; j < v; j++) {

if(adj[i][j] != 0) {

printf("%d -> %d : %d\n", i, j, adj[i][j]);

edges[e][0] = i;

edges[e][1] = j;

edges[e][2] = adj[i][j];

e++;

}

}

}

printf("Sorting...");

for(int i = 0; i < e; i++) {

for(int j = 0; j < e-i-1; j++) {

if(edges[j][2] > edges[j+1][2]) {

swapint(&edges[j][0], &edges[j+1][0]);

swapint(&edges[j][1], &edges[j+1][1]);

swapint(&edges[j][2], &edges[j+1][2]);

}

}

}

printf("\nThe sorted edges are : \n");

for(int i = 0; i < e; i++) {

printf("%d -> %d : %d\n", edges[i][0], edges[i][1], edges[i][2]);

}

for(int i = 0; i < v; i++) {

makeset(i);

}

for(int i = 0; i < e; i++) {

if(findset(edges[i][0]) != findset(edges[i][1])) {

mst[edges[i][0]][edges[i][1]] = edges[i][2];

unionp(edges[i][0], edges[i][1]);

}

}

int cost = 0;

printf("The MST is : \n");

for(int i = 0; i < v; i++) {

for(int j = i; j < v; j++) {

if(mst[i][j] != 0) {

printf("%d -> %d : %d\n", i, j, mst[i][j]);

cost += mst[i][j];

}

}

}

printf("Cost of MST is %d\n", cost);

return 0;

}

void makeset(int x) {

parent[x] = x;

rank[x] = 0;

}

int findset(int x) {

if(parent[x] != x) {

parent[x] = findset(parent[x]);

}

return parent[x];

}

void unionp(int x, int y) {

link(findset(x), findset(y));

}

void link(int a, int b) {

if(rank[a] > rank[b]) {

parent[b] = a;

}

else {

parent[a] = b;

if(rank[a] == rank[b])

rank[b]++;

}

}

void swapint(int\* a, int \*b) {

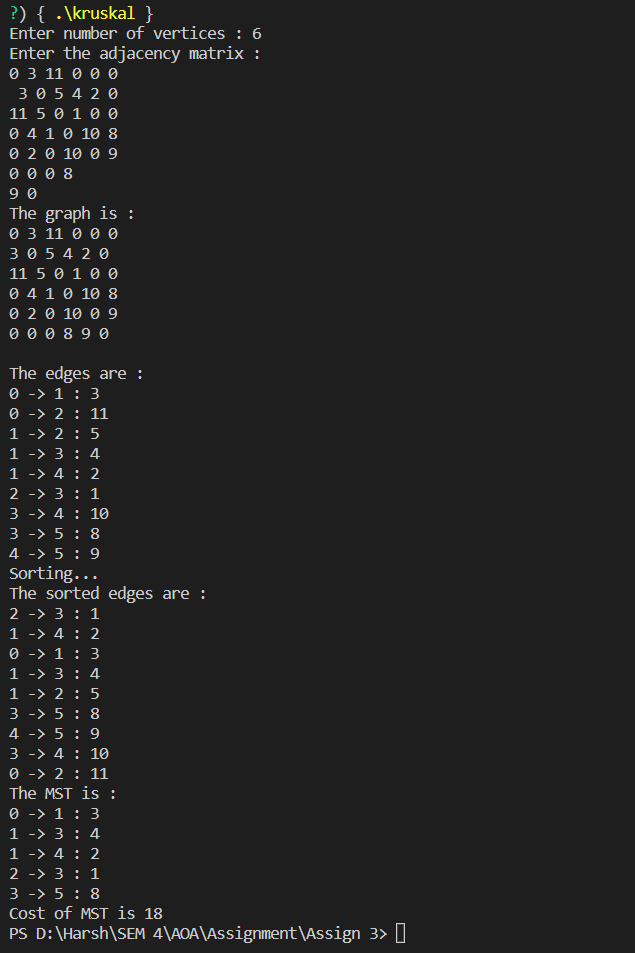
int temp = \*a;

\*a = \*b;

\*b = temp;

}

**OUTPUT:**

****