#include <stdio.h>

#include <stdlib.h>

int parent[10];

int find(int m) {

int p = m;

while (parent[p] != 0)

p = parent[p];

return p;

}

void unionSets(int i, int j) {

if (i < j)

parent[i] = j;

else

parent[j] = i;

}

void kruskal(int a[10][10], int n) {

int u = 0, v = 0, min, k = 0, i, j, sum = 0;

while (k < n - 1) {

min = 99;

for (i = 1; i <= n; i++) {

for (j = 1; j <= n; j++) {

if (a[i][j] < min && i != j) {

min = a[i][j];

u = i;

v = j;

}

}

}

i = find(u);

j = find(v);

if (i != j) {

unionSets(i, j);

printf("(%d,%d)=%d\n", u, v, a[u][v]);

sum += a[u][v];

k++;

}

a[u][v] = a[v][u] = 99;

}

printf("The cost of minimum spanning tree = %d\n", sum);

}

int main() {

int a[10][10];

int i, j, n;

printf("Enter the number of vertices of the graph: ");

scanf("%d", &n);

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

for (i = 1; i <= n; i++) {

for (j = 1; j <= n; j++) {

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

}

}

for (i = 1; i <= n; i++) {

parent[i] = 0;

}

kruskal(a, n);

return 0;

}

OUTPUT:

Enter the number of vertices of the graph: 3

Enter the weighted matrix:

0 1 3

1 0 2

3 2 0

(1,2)=1

(2,3)=2

The cost of minimum spanning tree = 3

=== Code Execution Successful ===