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(“\n\tImplementation of Kruskal’s Algorithm\n”);**

**Printf(“\nEnter the no. Of vertices:”);**

**Scanf(“%d”, & n);**

**Printf(“\nEnter 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(“\n\tMinimum 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) {**

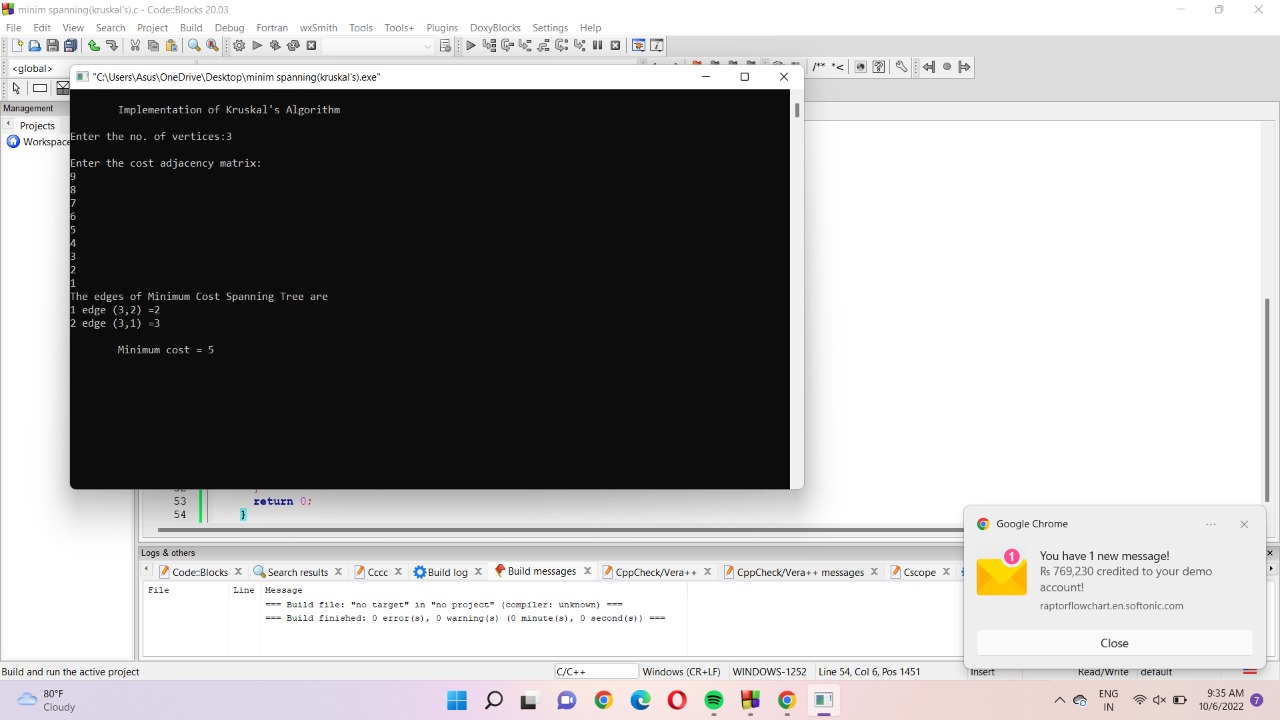
**Parent[j] = i;**

**Return 1;**

**}**

**Return 0;**

**}**

****