**5. Take an example subnet of hosts and obtain a broadcast tree for the subnet.**

**Program:**

#include <stdio.h>

int parent[20];

int find(int i) {

while (parent[i] != i)

i = parent[i];

return i;

}

void union\_set(int i, int j) {

int a = find(i);

int b = find(j);

parent[a] = b;

}

int main() {

int cost[20][20];

int n, i, j, edges = 1;

int min, u = -1, v = -1, mincost = 0;

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

scanf("%d", &n);

printf("Enter the cost adjacency matrix (use 99 for no connection):\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] = 99;

}

}

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

parent[i] = i;

printf("\nEdges in the Broadcast Tree:\n");

while (edges < n) {

min = 99;

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

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

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

min = cost[i][j];

u = i;

v = j;

}

}

}

union\_set(u, v);

printf("%d -> %d cost = %d\n", u, v, min);

mincost += min;

cost[u][v] = cost[v][u] = 99;

edges++;

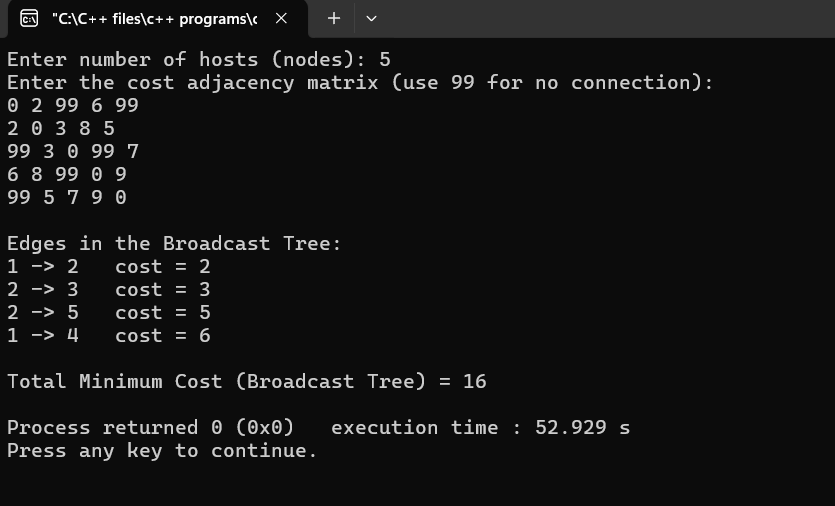
}

printf("\nTotal Minimum Cost (Broadcast Tree) = %d\n", mincost);

return 0;

}

**Output:**

****

**6. Implement distance vector routing algorithm for obtaining routing tables at each node.**

**Program:**

#include <stdio.h>

struct node {

int dist[10]; // Distance to other nodes

int from[10]; // From which node the packet will come

} rt[10];

int main() {

int costmat[10][10];

int nodes, i, j, k, count = 0;

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

scanf("%d", &nodes);

printf("Enter the cost matrix (999 for infinity):\n");

for (i = 0; i < nodes; i++) {

for (j = 0; j < nodes; j++) {

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

costmat[i][i] = 0;

rt[i].dist[j] = costmat[i][j];

rt[i].from[j] = j;

}

}

do {

count = 0;

for (i = 0; i < nodes; i++) {

for (j = 0; j < nodes; j++) {

for (k = 0; k < nodes; k++) {

if (rt[i].dist[j] > costmat[i][k] + rt[k].dist[j]) {

rt[i].dist[j] = costmat[i][k] + rt[k].dist[j];

rt[i].from[j] = k;

count++;

}

}

}

}

} while (count != 0);

printf("\nFinal Routing Tables:\n");

for (i = 0; i < nodes; i++) {

printf("\nRouter %d Table:\n", i + 1);

printf("Destination\tNext Hop\tDistance\n");

for (j = 0; j < nodes; j++) {

printf("%d\t\t%d\t\t%d\n", j + 1, rt[i].from[j] + 1, rt[i].dist[j]);

}

}

return 0;

}

**Output:**

