prims\_07.cpp

**Compile:** g++ prims\_07.cpp -o prims\_07

**Run:** ./prims\_07

**Program:**

#include<iostream>

#include<stdio.h>

using namespace std;

int n, cost[10][10];

void prim() {

inti, j, startVertex, endVertex;

int k, nr[10], temp, minimumCost = 0, tree[10][3];

temp = cost[0][0];

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

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

if (temp > cost[i][j]) {

temp = cost[i][j];

startVertex = i;

endVertex = j;

}

}

}

tree[0][0] = startVertex;

tree[0][1] = endVertex;

tree[0][2] = temp;

minimumCost = temp;

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

if (cost[i][startVertex] < cost[i][endVertex])

nr[i] = startVertex;

else

nr[i] = endVertex;

}

nr[startVertex] = 1000;

nr[endVertex] = 1000;

temp = 999;

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

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

if (nr[j] != 1000 && cost[j][nr[j]] < temp) {

temp = cost[j][nr[j]];

k = j;

}

}

tree[i][0] = k;

tree[i][1] = nr[k];

tree[i][2] = cost[k][nr[k]];

minimumCost = minimumCost + cost[k][nr[k]];

nr[k] = 1000;

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

if (nr[j] != 1000 && cost[j][nr[j]] > cost[j][k])

nr[j] = k;

}

temp = 999;

}

cout<<"\nThe min spanning tree is:-"<<endl;

for (i = 0; i< n - 1; i++) {

for (j = 0; j < 3; j++)

cout<<tree[i][j]<<"<<";

}

cout<<"\n";

cout<<"\nMin cost : " <<minimumCost<<endl;

}

int main()

{

inti, j;

cout<<"\nEnter the no. of vertices :"<<endl;

cin>>n;

cout<<"\nEnter the costs of edges in matrix form :";

for (i = 0; i< n; i++)

{

for (j = 0; j < n; j++)

{

cout<<"Enter the cost of edge "<<i<<j<<" : "<<endl;

cin>>cost[i][j];

}

}

cout<<"\nThe matrix is : "<<endl;

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

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

cout<<"\t"<<cost[i][j];

}

cout<<"\n";

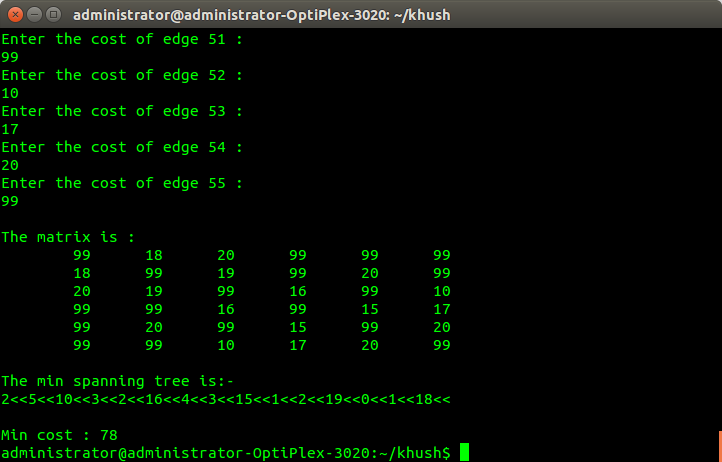
}

prim();

return 0;

}

**Output:**

****