**Assignment No:  8**

**Problem Statement:**

Represent any real-world graph using adjacency list /adjacency matrix find minimum spanning tree using Prim‟s algorithm.

**Outcomes:**

Students will be able to:

1. Make use of coding standards for application development.

**Code:**

#include <bits/stdc++.h>

using namespace std;

#define V 5

bool createsMST(int u, int v, vector<bool> inMST){

if (u == v)

return false;

if (inMST[u] == false && inMST[v] == false)

return false;

else if (inMST[u] == true && inMST[v] == true)

return false;

return true;

}

void printMinSpanningTree(int cost[][V]){

vector<bool> inMST(V, false);

inMST[0] = true;

int edgeNo = 0, MSTcost = 0;

while (edgeNo < V - 1) {

int min = INT\_MAX, a = -1, b = -1;

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

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

if (cost[i][j] < min) {

if (createsMST(i, j, inMST)) {

min = cost[i][j];

a = i;

b = j;

}

}

}

}

if (a != -1 && b != -1) {

cout<<"Edge "<<edgeNo++<<" : ("<<a<<" , "<<b<<" ) : cost = "<<min<<endl;

MSTcost += min;

inMST[b] = inMST[a] = true;

}

}

cout<<"Cost of Minimum spanning tree ="<<MSTcost;

}

int main() {

int cost[][V] = {

{ INT\_MAX, 12, INT\_MAX, 25, INT\_MAX },

{ 12, INT\_MAX, 11, 8, 12 },

{ INT\_MAX, 11, INT\_MAX, INT\_MAX, 17 },

{ 25, 8, INT\_MAX, INT\_MAX, 15 },

{ INT\_MAX, 12, 17, 15, INT\_MAX },

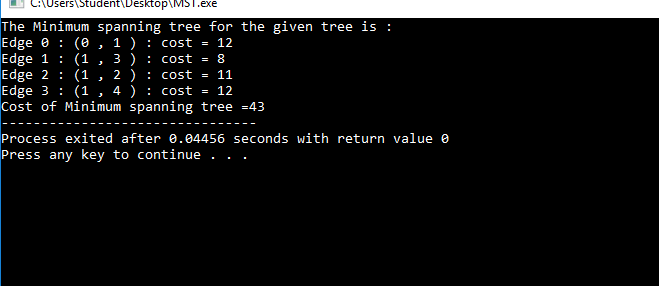
};

cout<<"The Minimum spanning tree for the given tree is :\n";

printMinSpanningTree(cost);

return 0;

}

**Output**