**DSA LAB – 11**

**Name:** Etcherla Sai Manoj **Mis. No:** 112015044 **Branch:** CSE

**Question 1:**

**Code:**

#include<bits/stdc++.h>

using namespace std;

//define number of vertices of graph

#define vertex 7

int main () {

// create a 2D array of size (vetex \* vertex)

// This array is adjacency matrix representation of graph

int graph[vertex][vertex] = {

{0,28,0,0,0,10,0},

{28,0,16,0,0,0,14},

{0,16,0,12,0,0,0},

{0,0,12,22,0,18},

{0,0,0,22,0,25,24},

{10,0,0,0,25,0,0},

{0,14,0,18,24,0,0}

};

// number of edge declaration

int line = 0;

// an array declaration to keep checking on if a vertex is visited or not

int node\_visited[vertex];

// the array is bool type all elements declared to false

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

node\_visited[i]=false;

}

// the element becomes true if the vertex is visited

node\_visited[0] = true;

int row, column;

cout << "------------------\n";

cout << "Edge\t : Weight\n";

cout << "------------------\n";

while (line < vertex - 1){

int minimum = INT\_MAX;

row = 0, column = 0;

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

if (node\_visited[i]){

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

if (!node\_visited[j] && graph[i][j]){

if (minimum > graph[i][j]) {

minimum = graph[i][j];

row = i;

column = j;

}

}

}

}

}

cout << row << " ---> " << column << " : " << graph[row][column] << endl;

node\_visited[column] = true;

line++;

}

cout << "------------------\n";

return 0;

}

**Input & Output:**

