# Practical 7 Source Code:-

#include <iostream> #include <cstring>

using namespace std;

#define INF 9999999

#define V 5 // Define the number of vertices

// Create a 2D array of size 5x5 for the adjacency matrix to represent the graph int G[V][V] = {

{0, 9, 75, 0, 0},

{9, 0, 95, 19, 42},

{75, 95, 0, 51, 66},

{0, 19, 51, 0, 31},

{0, 42, 66, 31, 0}

};

int main() { // Number of edges

int no\_edge = 0;

// Create an array to track selected vertices bool selected[V];

memset(selected, false, sizeof(selected)); // Set all to false

// Set the first vertex as selected selected[0] = true;

int x, y; // Row and column numbers for edges

// Print for edge and weight

cout << "Edge : Weight" << endl;

// Loop until we include V-1 edges in the MST

while (no\_edge < V - 1) { int min = INF; x = 0; y = 0;

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

if (selected[i]) { // If vertex i is selected

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

// Check for edges from selected vertex to unselected vertex if (!selected[j] && G[i][j]) {

// If there's an edge and it's weight is less than min if (min > G[i][j]) { min = G[i][j];

1. = i; // Store selected vertex
2. = j; // Store unselected vertex

}

}

}

}

}

// Print the selected edge

cout << x << " - " << y << " : " << G[x][y] << endl;

// Include the selected vertex in the MST selected[y] = true; no\_edge++;

}

return 0;

}

# Output:-

