#include <stdio.h>

#include <limits.h>

#define MAX\_VERTICES 100

struct Graph {

int numVertices;

int adjMatrix[MAX\_VERTICES][MAX\_VERTICES];

};

void initializeGraph(struct Graph\* graph, int numVertices) {

graph->numVertices = numVertices;

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

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

graph->adjMatrix[i][j] = 0; // Initialize with no edges

}

}

}

void addEdge(struct Graph\* graph, int src, int dest, int weight) {

graph->adjMatrix[src][dest] = weight;

graph->adjMatrix[dest][src] = weight; // For undirected graph

}

int minKey(int key[], int mstSet[], int numVertices) {

int min = INT\_MAX, minIndex;

for (int v = 0; v < numVertices; v++) {

if (!mstSet[v] && key[v] < min) {

min = key[v];

minIndex = v;

}

}

return minIndex;

}

void primMST(struct Graph\* graph) {

int parent[MAX\_VERTICES]; // To store the constructed MST

int key[MAX\_VERTICES]; // Key values used to pick minimum weight edge

int mstSet[MAX\_VERTICES]; // To represent set of vertices included in MST

for (int i = 0; i < graph->numVertices; i++) {

key[i] = INT\_MAX;

mstSet[i] = 0;

}

key[0] = 0; // Start from the first vertex

parent[0] = -1; // First vertex is the root of MST

for (int count = 0; count < graph->numVertices - 1; count++) {

int u = minKey(key, mstSet, graph->numVertices);

mstSet[u] = 1;

for (int v = 0; v < graph->numVertices; v++) {

if (graph->adjMatrix[u][v] && !mstSet[v] &&

graph->adjMatrix[u][v] < key[v]) {

parent[v] = u;

key[v] = graph->adjMatrix[u][v];

}

}

}

printf("Edge Weight\n");

for (int i = 1; i < graph->numVertices; i++) {

printf("%d - %d %d\n", parent[i], i, graph->adjMatrix[i][parent[i]]);

}

}

int main() {

struct Graph graph;

int numVertices, numEdges;

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

scanf("%d", &numVertices);

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

scanf("%d", &numEdges);

initializeGraph(&graph, numVertices);

printf("Enter edges and weights (source destination weight):\n");

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

int src, dest, weight;

scanf("%d %d %d", &src, &dest, &weight);

addEdge(&graph, src, dest, weight);

addEdge(&graph, dest, src, weight); // For undirected graph

}

primMST(&graph);

return 0;

}

OUTPUT:

