

```

#include<iostream>
#include<climits>
using namespace std;

int findMinVertex(int* weights, bool* visited, int n){

    int minVertex = -1;
    for(int i = 0; i < n; i++){
        if(!visited[i] && (minVertex == - 1 || weights[i] < weights[minVertex])){
            minVertex = i;
        }
    }
    return minVertex;
}

```

```

void prims(int** edges, int n){

    int* parent = new int[n];
    int* weights = new int[n];
    bool* visited = new bool[n];

    for(int i = 0; i < n; i++){
        visited[i] = false;
        weights[i] = INT_MAX;
    }
    parent[0] = -1;
    weights[0] = 0;

    for(int i = 0; i < n - 1; i++){
        // Find Min Vertex
        int minVertex = findMinVertex(weights, visited, n);
        visited[minVertex] = true;
        // Explore un visted neighbours
        for(int j = 0; j < n; j++){
            if(edges[minVertex][j] != 0 && !visited[j]){
                if(edges[minVertex][j] < weights[j]){
                    weights[j] = edges[minVertex][j];
                    parent[j] = minVertex;
                }
            }
        }
    }

    for(int i = 1; i < n; i++){
        if(parent[i] < i){
            cout << parent[i] < " << i << " " << weights[i] << endl;
        }else{

```

```

        cout << i << " " << parent[i] << " " << weights[i] << endl;
    }
}

int main() {
    int n;
    int e;
    cin >> n >> e;
    int** edges = new int*[n];
    for (int i = 0; i < n; i++) {
        edges[i] = new int[n];
        for (int j = 0; j < n; j++) {
            edges[i][j] = 0;
        }
    }

    for (int i = 0; i < e; i++) {
        int f, s, weight;
        cin >> f >> s >> weight;
        edges[f][s] = weight;
        edges[s][f] = weight;
    }
    cout << endl;
    prims(edges, n);

    for (int i = 0; i < n; i++) {
        delete [] edges[i];
    }
    delete [] edges;
}

```