#include <iostream>

#include <vector>

#include <limits>

#include <queue>

using namespace std;

const int INF = numeric\_limits<int>::max();

//رغد دباغ , جودي مغربي , سيدرا حاج محمود

struct Edge {

    int to, weight;

};

//رغد دباغ

vector<int> cheriyanMehlhornGabow(int vertices, const vector<vector<Edge>>& graph, int source) {

    vector<int> distance(vertices, INF);

    distance[source] = 0;

    //سيدرا حاج محمود

    priority\_queue<pair<int, int>, vector<pair<int, int>>, greater<pair<int, int>>> pq;

    pq.push({0, source});

    while (!pq.empty()) {

        auto [dist, u] = pq.top();

        pq.pop();

        //سيدرا حاج محمود

        if (dist > distance[u]) continue;

        //جودي مغربي

        for (const Edge& edge : graph[u]) {

            int v = edge.to;

            int weight = edge.weight;

            //جودي مغربي

            if (distance[u] + weight < distance[v]) {

                distance[v] = distance[u] + weight;

                pq.push({distance[v], v});

            }

        }

    }

    return distance;

}

//رغد دباغ

int main() {

    int vertices, edges;

    cout << "أدخل عدد الرؤوس: ";

    cin >> vertices;

    cout << "أدخل عدد الحواف: ";

    cin >> edges;

    vector<vector<Edge>> graph(vertices);

    cout << "أدخل الحواف (من، إلى، الوزن):\n";

    for (int i = 0; i < edges; i++) {

        int from, to, weight;

        cin >> from >> to >> weight;

        graph[from].push\_back({to, weight});

    }

    int source;

    cout << "أدخل رأس المصدر: ";

    cin >> source;

    vector<int> distances = cheriyanMehlhornGabow(vertices, graph, source);

    cout << "المسافات الأقصر من الرأس " << source << ":\n";

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

        if (distances[i] == INF) {

            cout << "إلى الرأس " << i << ": غير متصل\n";

        } else {

            cout << "إلى الرأس " << i << ": " << distances[i] << "\n";

        }

    }

    return 0;

}

//رغد دباغ , سيدرا حاج محمود , جودي مغربي