//#include <iostream>

//#include <vector>

//using namespace std;

//

//typedef vector< vector<float>> graph;

//

//graph buildGraph(int numVertices);

//void addEdge(graph &g, float w, int r, int c);

//void removeEdge(graph &g, int r, int c);

//void printGraph(graph g);

//

//struct ledge {

// int v;

// float w;

//};

//typedef vector<vector <ledge>> lgraph;

//lgraph buildlGraph(int )

//

//

//

//void addEdge(graph& g, float w, int r, int c) {

// if (r < 0 || r >= g.size() || (c < 0 || c >= g.size())) {

// return;

// }

// g[r][c] = w;

//}

//

//struct edge{

// int u, v;

// float w;

//};

//

//void addEdge(graph& g, edge e) {

// if (e.u < 0 || e.u >= g.size() || (e.v < 0 || e.v >= g.size())) {

// return;

// }

// g[e.u][e.v] = e.w;

//

//}

//

//graph buildGraph(int numVertices) {

// graph g(numVertices);

//

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

// g[i] = vector<float>(numVertices, 0);

// }

// return g;

//}

//

//

//

//int main() {

//

// int n = 9;

// graph g = buildGraph(6);

//

//

//

//

// return 0;

//}