class Main {

public static void main(String[] args) {

Graph graph = new Graph(4);

graph.addEdge(0, 1);

graph.addEdge(0, 2);

graph.addEdge(0, 3);

graph.addEdge(1, 2);

graph.printGraph();

}

}

class Graph {

int[][] graph;

int vertices;

Graph(int vertices) {

graph = new int[vertices][vertices];

this.vertices = vertices;

}

void addEdge(int src, int dest) {

if (src >= 0 && dest >= 0 && src < vertices && dest < vertices && src != dest) {

graph[src][dest] = 1;

graph[dest][src] = 1;

} else {

System.out.println("Invalid source and destination.");

}

}

void removeEdge(int src, int dest) {

if (src >= 0 && dest >= 0 && src < vertices && dest < vertices && src != dest) {

graph[src][dest] = 0;

graph[dest][src] = 0;

} else {

System.out.println("Invalid source and destination.");

}

}

boolean isValid(int src, int dest) {

return src >= 0 && dest >= 0 && src < vertices && dest < vertices && src != dest;

}

boolean havingEdge(int src, int dest) {

if (isValid(src, dest)) {

return graph[src][dest] == 1;

}

System.err.println("Invalid source and destination.");

return false;

}

void printGraph() {

System.out.println("Matrix representation of graph:");

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

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

System.out.print(graph[i][j] + " ");

}

System.out.println();

}

}

}