import java.util.Arrays;  
import java.util.Scanner;  
public class Main {  
 private static final int *INF* = Integer.*MAX\_VALUE*;  
 public static void dijkstra(int[][] graph, int source) {  
 int n = graph.length;  
 int[] dist = new int[n];  
 int[] pred = new int[n];  
 boolean[] visited = new boolean[n];  
 Arrays.*fill*(dist, *INF*);  
 Arrays.*fill*(pred, -1);  
 dist[source] = 0;  
 for (int i = 0; i < n - 1; i++) {  
 int u = *minDistance*(dist, visited);  
 visited[u] = true;  
 for (int v = 0; v < n; v++) {  
 if (!visited[v] && graph[u][v] != 0 && dist[u] != *INF* &&  
 dist[u] + graph[u][v] < dist[v]) {  
 dist[v] = dist[u] + graph[u][v];  
 pred[v] = u;  
 }  
 }  
 }  
 *printShortestPaths*(source, dist, pred);  
 }  
 private static int minDistance(int[] dist, boolean[] visited) {  
 int minDist = *INF*;  
 int minVertex = -1;  
 for (int i = 0; i < dist.length; i++) {  
 if (!visited[i] && dist[i] < minDist) {  
 minDist = dist[i];  
 minVertex = i;  
 }  
 }  
 return minVertex;  
 }  
 private static void printShortestPaths(int source, int[] dist, int[] pred)  
 {  
 System.*out*.println("Shortest paths from the source:");  
 for (int i = 0; i < dist.length; i++) {  
 if (i != source) {  
 if (dist[i] == *INF*) {  
 System.*out*.println("No path from " + source + " to " + i);  
 } else {  
 System.*out*.print("Shortest path from " + source + " to " +  
 i + ": ");  
 *printPath*(source, i, pred);  
 System.*out*.println(" (Distance: " + dist[i] + ")");  
 }  
 }  
 }  
// Print the final shortest path  
 System.*out*.print("Final shortest path from " + source + " to all vertices: ");  
 for (int i = 0; i < dist.length; i++) {  
 if (i != source) {  
 *printPath*(source, i, pred);  
 System.*out*.print(" (Distance: " + dist[i] + ")");  
 System.*out*.println();  
 }  
 }  
 }  
 private static void printPath(int source, int dest, int[] pred) {  
 if (source == dest) {  
 System.*out*.print(source + " ");  
 } else if (pred[dest] == -1) {  
 System.*out*.println("No path found");  
 } else {  
 *printPath*(source, pred[dest], pred);  
 System.*out*.print(dest + " ");  
 }  
 }  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter the number of vertices: ");  
 int n = scanner.nextInt();  
 int[][] graph = new int[n][n];  
 System.*out*.println("Enter the adjacency matrix:");  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < n; j++) {  
 graph[i][j] = scanner.nextInt();  
 }  
 }  
 System.*out*.print("Enter the source vertex: ");  
 int source = scanner.nextInt();  
 *dijkstra*(graph, source);  
 scanner.close();  
 }  
}