```java
import java.util.Scanner;
public class Main {
    static final int INF = 9999;
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        System.out.println("Enter number of nodes: ");
        int n = scn.nextInt();
        scn.nextLine();
        char[] nod = new char[n];
        for (int i = 0; i < n; i++) {
            nod[i] = (char) ('A' + i);
        }
        int[][] dis = new int[n][n];
        int[][] hop = new int[n][n];
        System.out.println("Enter the distances (9999 if no connection): ");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                System.out.print("Distance from " + nod[i] + " to " + nod[j] + ": ");
                dis[i][j] = scn.nextInt();
                if (i == j) {
                    dis[i][j] = 0;
                }
                hop[i][j] = (dis[i][j] == INF) ? -1 : j;
            }
        }
        System.out.println("\nInitial Tables:");
        dspTbl(nod, dis, hop);
        for (int iter = 0; iter < n - 2; iter++) {
            for (int i = 0; i < n; i++) { // for each node
```

```java
            for (int j = 0; j < n; j++) { // check distance to every other node
                for (int k = 0; k < n; k++) { // compare via every other node
                    if (dis[i][k] + dis[k][j] < dis[i][j]) {
                        dis[i][j] = dis[i][k] + dis[k][j];
                        hop[i][j] = hop[i][k];
                    }
                }
            }
        }
        System.out.println("\nAfter iteration " + (iter + 1) + ":");
        dspTbl(nod, dis, hop);
    }
    System.out.println("\nFinal Tables:");
    dspTbl(nod, dis, hop);
}
private static void dspTbl(char[] nod, int[][] dis, int[][] hop) {
    int n = nod.length;
    for (int i = 0; i < n; i++) {
        System.out.println("Table for node " + nod[i] + ":");
        System.out.println("Dst\tDis\tHop");
        for (int j = 0; j < n; j++) {
            if (dis[i][j] == INF) {
                System.out.println(nod[j] + "\tINF\t-");
            } else {
                System.out.println(nod[j] + "\t" + dis[i][j] + "\t" + nod[hop[i][j]]);
            }
        }
        System.out.println();
    }}}
```