

COP 3503 Recitation #9 Problem: Floyd and Warshall (Week of 7/15)
Due: 7/26/13 (Thursday) at 11:55 PM Webcourses2 time

Write complete Java program. Please include ample comments in your code. Using Big-O notation, indicate the time complexity in terms of the appropriate variables.

For this recitation, you will implement algorithms 8.5.4 and 8.5.5 from textbook (see pages: 354-355).

The program will take file input, from the file 'input.txt'. The general form of the file will be as follows.

```
<graph representation>
...
...
<graph representation>
0
```

<Graph representation> will be an adjacency matrix. The 0 at the end is used to indicate end of input.

An adjacency matrix is denoted as follows. The first line will give an integer N , indicating the number of vertices in the graph. This is followed by N lines, each containing N integers ("-" if there is no direct path between two vertices) – basically a straightforward representation of the adjacency matrix.

For output, your program must compute the length of a shortest path between each pair of vertices and print the shortest paths. You should write the result to standard output. The format of the output will match the matrix and list format described above. Note that you must include the first line that gives the number of vertices, just as in the input. Please see the next page for sample input/output.

Sample Input

```
5
0 8 3 1 -
8 0 4 - 2
3 4 0 1 1
1 - 1 0 8
- 2 1 8 0
0
```

Sample Output

```
4
Lengths of shortest paths:
0 5 2 1 3
5 0 3 4 2
2 3 0 1 1
1 4 1 0 2
3 2 1 2 0
Shortest paths:
(1 1) (1 4 3 5 2) (1 4 3) (1 4) (1 4 3 5)
(2 5 3 4 1) (2 2) (2 5 3) (2 5 3 4) (2 5)
(3 4 1) (3 5 2) (3 3) (3 4) (3 5)
(4 1) (4 3 5 2) (4 3) (4 4) (4 3 5)
(5 3 4 1) (5 2) (5 3) (5 3 4) (5 5)
```

Note: Directly read your input from the input file, input.in. Turn in your program file, Floyd.java.