**package** AlgorithmPresentation;

**import** java.io.File;

**import** java.util.Scanner;

**class** allPairsShortestPath

{

**public int**[][] **distance**;

**public int vertex**;

**public int**[][] **finalDistance**;

**public** allPairsShortestPath(**int**[][] distance, **int** vertex) {

**this**.**distance** = distance;

**this**.**vertex** = vertex;

**this**.**finalDistance** = **new int**[vertex+1][vertex+1];

}

**public void** initialDistance(){

**for** (**int** i=1;i<=**vertex**;i++){

**for** (**int** j=1;j<=**vertex**;j++){

**if** (i==j){

**distance**[i][j] = 0;

}

**else**{

**distance**[i][j] = **distance**[i][j];

}

}

}

System.***out***.println(**"-----------------------------------------------------------------------"**);

**for** (**int** i=1;i<=**vertex**;i++){

**for** (**int** j = 1; j <=**vertex**; j++){

**finalDistance**[i][j] = **distance**[i][j];

}

}

**for** (**int** k=1;k<=**vertex**;k++) {

**for** (**int** i=1;i<=**vertex**;i++) {

**for** (**int** j=1;j<=**vertex**;j++) {

**int** distance = **finalDistance**[i][k] + **finalDistance**[k][j];

**finalDistance**[i][j] = Math.*min*(distance,**finalDistance**[i][j]);

}

}

}

System.***out***.println(**"The result of (all pairs shortest path using Floyd Warshall Algorithm):"**);

**for** (**int** i=1;i<=**vertex**;i++) {

**for** (**int** j = 1; j <=**vertex** ; j++) {

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

}

System.***out***.println();

}

System.***out***.println(**"-----------------------------------------------------------------------"**);

}

}

**public class** floydWarshall {

**public static void** main(String[] args) {

File file = **new** File(**"C:/Users/Hp/Desktop/presentation.txt"**);

**int**[][] graph = **new int**[10][10];

**int** vertex = 0;

**if** (file.exists()) {

**try** {

Scanner input = **new** Scanner(file);

**int** count = 0;

**while** (input.hasNext()) {

String data = input.nextLine();

String split[] = data.split(**" "**);

**if** (count==0) {

**for** (**int** i=1;i<=Integer.*parseInt*(split[0]);i++) {

**for** (**int** j = 1; j <=Integer.*parseInt*(split[0]) ; j++) {

graph[i][j] = (Integer.***MAX\_VALUE***/2);

}

}

count = count + 1;

vertex = Integer.*parseInt*(split[0]);

}

**else** {

graph[Integer.*parseInt*(split[0])][Integer.*parseInt*(split[1])] = Integer.*parseInt*(split[2]);

}

}

}**catch** (Exception e) {

System.***out***.println(e.getMessage());

}

}

allPairsShortestPath object = **new** allPairsShortestPath(graph,vertex);

object.initialDistance();

}

}