import java.io.\*;import java.util.\*;

public class Lab4TSP {

final static int CITIES = 12;

static int [][] adjacency = new int[CITIES][CITIES];

static int tempCost = 0;

static int bestCost = 9000;

public static void main(String[] args){

try{

populateMatrix(adjacency);

outputMatrix(adjacency);

}//try

catch(Exception e){

System.out.println("Somthing went wrong");

}//catch

ArrayList <Integer> partialTour = new ArrayList<>();

ArrayList <Integer> remainingCities = new ArrayList<>();

partialTour.add(0);

for(int i = 1; i < CITIES; i++){

remainingCities.add(i);

}//for

long startTime = System.nanoTime();

recTSP(partialTour,remainingCities);

System.out.println("Best Cost:" + bestCost);

long elapsedTime = System.nanoTime() - startTime;

System.out.printf("Time: %3.10f\n",elapsedTime/(Math.pow(10, 9)));

}//main

public static void populateMatrix(int[][] adj)throws FileNotFoundException{

File f = new File("tsp12.txt");

Scanner input = new Scanner(f);

int value;

for(int i = 0; i < CITIES && input.hasNext(); i++){

for(int j=i; j<CITIES && input.hasNext();j++){

if(i==j)

adj[i][j]=0;

else{

value=input.nextInt();

System.out.print(value+":");

adj[i][j]=value;

adj[j][i]=value;

}//else

}//for j

}//for i

System.out.println("===============================\n\n");

input.close();

}//populate function

public static void outputMatrix(int[][] adj){

for(int i=0;i<adj.length;i++){

for(int j=0;j<adj.length;j++){

System.out.print(adj[i][j]+"\t");

}//for j

System.out.println("");

}//for i

System.out.println("===============================\n\n");

}//output function

public static int computeCost(ArrayList<Integer>tour){

int totalCost=0;

for(int i=0;i<tour.size()-1;i++){

totalCost+=adjacency[tour.get(i)][tour.get(i+1)];

}//for

if(tour.size()==CITIES)

totalCost+=adjacency[tour.get(tour.size()-1)][0];

return totalCost;

}//computeCost

public static void recTSP(ArrayList<Integer> partialTour,ArrayList<Integer>remainingCities){

if(remainingCities.isEmpty()){

tempCost=computeCost(partialTour);

if(tempCost<bestCost){

bestCost=tempCost;

System.out.println("Current best tour: ");

for(int i=0;i<partialTour.size();i++){

System.out.print((char) (65+partialTour.get(i))+" ");

}//for

System.out.println("Total Cost: "+tempCost);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}//if

}//if

else{

for(int i=0;i<remainingCities.size();i++){

ArrayList<Integer> newPartialTour=(ArrayList)(partialTour.clone());

newPartialTour.add(remainingCities.get(i));

tempCost=computeCost(newPartialTour);

if(tempCost<bestCost){

ArrayList <Integer> newRemainingCities=(ArrayList)(remainingCities.clone());

newRemainingCities.remove(i);

recTSP(newPartialTour,newRemainingCities);

}//if

else{

break;

}//else

}//for

}//else

}//recTSP

}//class

//OUTPUT   
/\*

Current best tour:

A F D J K I E C B L G H Total Cost: 821

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Best Cost:821

Time: 0.4386592000  
\*/