**ASSIGNMENT 8**

**import java.util.Scanner;**

**class tsp1 {**

**public StringBuilder path;**

**public int tsp(int mask, int pos, int dist[][], int n) {**

**int visited = (1 << n) - 1;**

**if (mask == visited)**

**return dist[pos][0];**

**int ans = Integer.MAX\_VALUE;**

**int bestCity = -1;**

**StringBuilder bestPath = new StringBuilder();**

**for (int city = 0; city < n; city++) {**

**if ((mask & (1 << city)) == 0) {**

**int newAns = dist[pos][city] + tsp((mask | (1 << city)), city, dist, n);**

**if (newAns < ans) {**

**ans = newAns;**

**bestCity = city;**

**bestPath = new StringBuilder(path);**

**bestPath.append(" -> ").append(city);**

**}**

**}**

**}**

**if (bestCity != -1) {**

**path = bestPath;**

**}**

**return ans;**

**}**

**}**

**public class Main {**

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

**Scanner sc = new Scanner(System.in);**

**System.out.print("Enter number of cities: ");**

**int n = sc.nextInt();**

**int dist[][] = new int[n][n];**

**System.out.println("Enter distance matrix (enter each element on a new line):");**

**for (int i = 0; i < n; i++) {**

**for (int j = 0; j < n; j++) {**

**dist[i][j] = sc.nextInt();**

**}**

**}**

**tsp1 t = new tsp1();**

**t.path = new StringBuilder("0");**

**System.out.println("MinCost: " + t.tsp(1, 0, dist, n));**

**System.out.println(t.path.toString());**

**}**

**}**

**OUTPUT:**

Enter number of cities: 4

Enter distance matrix (enter each element on a new line):

0

20

42

25

20

0

30

34

42

30

0

10

25

34

10

0

MinCost: 85

0 -> 3 -> 2 -> 1