**Minimum Transport Cost**

**Time Limit: 2000/1000 MS (Java/Others)    Memory Limit: 65536/32768 K (Java/Others)  
Total Submission(s): 9757    Accepted Submission(s): 2614**

Problem Description

These are N cities in Spring country. Between each pair of cities there may be one transportation track or none. Now there is some cargo that should be delivered from one city to another. The transportation fee consists of two parts:   
The cost of the transportation on the path between these cities, and  
  
a certain tax which will be charged whenever any cargo passing through one city, except for the source and the destination cities.  
  
You must write a program to find the route which has the minimum cost.

Input

First is N, number of cities. N = 0 indicates the end of input.  
  
The data of path cost, city tax, source and destination cities are given in the input, which is of the form:  
  
a11 a12 ... a1N  
a21 a22 ... a2N  
...............  
aN1 aN2 ... aNN  
b1 b2 ... bN  
  
c d  
e f  
...  
g h  
  
where aij is the transport cost from city i to city j, aij = -1 indicates there is no direct path between city i and city j. bi represents the tax of passing through city i. And the cargo is to be delivered from city c to city d, city e to city f, ..., and g = h = -1. You must output the sequence of cities passed by and the total cost which is of the form:

Output

From c to d :  
Path: c-->c1-->......-->ck-->d  
Total cost : ......  
......  
  
From e to f :  
Path: e-->e1-->..........-->ek-->f  
Total cost : ......  
  
Note: if there are more minimal paths, output the lexically smallest one. Print a blank line after each test case.

Sample Input

5

0 3 22 -1 4

3 0 5 -1 -1

22 5 0 9 20

-1 -1 9 0 4

4 -1 20 4 0

5 17 8 3 1

1 3

3 5

2 4

-1 -1

0

Sample Output

From 1 to 3 :

Path: 1-->5-->4-->3

Total cost : 21

From 3 to 5 :

Path: 3-->4-->5

Total cost : 16

From 2 to 4 :

Path: 2-->1-->5-->4

Total cost : 17

分析：先给你一个n\*n的矩阵 如果map[i][j] = -1,则代表i到j没有路径 然后再给你n个城市要收取的税（经过每个城市都要收取一个的费用） 让你找从A到B的最少费用 用Floyd较好 判断的时候多加一个税收的费用

注意：在满足费用最小的情况下 若有多条路径 则要输出字典序 是最小的

AC代码：

#include <iostream>

#include <cstdio>

#define INF 0x3f3f3f3f

#define MAX 100

using namespace std;

int n,A,B;

int map[MAX][MAX],path[MAX][MAX],a[MAX];

void init()

{

for(int i = 1; i <= n; i++)

{

for(int j = 1; j <= n; j++)

{

scanf("%d",&map[i][j]);

if(map[i][j]==-1)

map[i][j] = INF;

path[i][j] = j; ///保存最短路径 存路径的下一个点

}

}

}

int Floyd()

{

for(int k = 1; k <= n; k++)

for(int i = 1; i <= n; i++)

for(int j = 1; j <= n; j++)

{

if(map[i][j] > map[i][k] + map[k][j] + a[k]) ///没过一个点 就要把这个点的值加在一起比较一下

{

map[i][j] = map[i][k] + map[k][j] + a[k];

path[i][j] = path[i][k];

}

if(map[i][j] == map[i][k] + map[k][j] + a[k])

{

if(path[i][j] > path[i][k])

path[i][j] = path[i][k];

}

}

}

void put\_path(int x,int y) ///输出最短路径 先输出第一个 再找下一个 直到最后一个

{

int k;

if(x==y)

{

printf("%d\n",y);

return;

}

k = path[x][y];

printf("%d-->",x);

put\_path(k,y);

}

int main()

{

while(~scanf("%d",&n))

{

if(n==0)

break;

init();

for(int i = 1; i <= n; i++)

scanf("%d",&a[i]);

Floyd();

while(~scanf("%d%d",&A,&B))

{

if(A==-1&&B==-1)

break;

if(A==B)

{

printf("From %d to %d :\n",A,B);

printf("Path: %d\n",A);

printf("Total cost : %d\n\n",map[A][B]);

continue;

}

printf("From %d to %d :\n",A,B);

printf("Path: ");

put\_path(A,B);

printf("Total cost : %d\n\n",map[A][B]);

}

}

return 0;

}