#include<iostream>

#include<stack>

using namespace std;

#define MAX 1000

int Graph[MAX][MAX];

int Dis[MAX][MAX];

#define infinite 1000

int path[MAX][MAX];

void floyd(int N)

{

int i,j,k;

for(k=0;k<N;k++)

{

for(i=0;i<N;i++)

{

for(j=0;j<N;j++)

{

if(Dis[i][k]+Dis[k][j]<Dis[i][j])

{

Dis[i][j]=Dis[i][k]+Dis[k][j];

path[i][j]=k;

}

}

}

}

}

void print\_path(int N) //打印路径

{

int i,j;

for(i=0;i<N;i++)

{

for(j=0;j<N;j++)

{

if((i!=j) &&Dis[i][j]!=infinite)

{

cout<<i+1<<"----"<<j+1<<" distance:"<<Dis[i][j]<<endl;

cout<<"path:"<<endl;

int k=j;

stack <int> ph;

do

{

k=path[i][k];

ph.push(k);

}while(k!=i);

cout<<ph.top()+1;

ph.pop();

while(!ph.empty())

{

cout<<"->"<<ph.top()+1;

ph.pop();

}

cout<<"->"<<j+1<<endl;

}

}

}

}

int main()

{

int N,i,j;

cin>>N;

for(i=0;i<N;i++)

{

for(j=0;j<N;j++)

{

int g;

cin>>g;

Graph[i][j]=g;

Dis[i][j]=g;

}

}

//初始化路径

for(i=0;i<N;i++)

{

for(j=0;j<N;j++)

{

path[i][j]=i;

}

}

floyd(N);

print\_path(N);

}