2006/2007 ACM International Collegiate Programming Contest   
University of Ulm Local Contest

# Problem H: Homogeneous squares

Source file: homogeneous.(c|cc|hs|java|pas)   
Input file: homogeneous.in

Assume you have a square of size *n* that is divided into *n×n* positions just as a checkerboard. Two positions *(x1,y1)* and *(x2,y2)*, where *1 ≤ x1,y1,x2,y2 ≤ n*, are called "independent" if they occupy different rows and different columns, that is, *x1≠x2* and *y1≠y2*. More generally, *n* positions are called independent if they are pairwise independent. It follows that there are *n!* different ways to choose *n* independent positions.

Assume further that a number is written in each position of such an *n×n* square. This square is called "homogeneous" if the sum of the numbers written in *n* independent positions is the same, no matter how the positions are chosen. Write a program to determine if a given square is homogeneous!

**Input Specification**

The input contains several test cases.   
The first line of each test case contains an integer *n* (*1 ≤ n ≤ 1000*). Each of the next *n* lines contains *n* numbers, separated by exactly one space character. Each number is an integer from the interval *[-1000000,1000000]*.   
The last test case is followed by a zero.

#### Output Specification

For each test case output whether the specified square is homogeneous or not. Adhere to the format shown in the sample output.

#### Sample Input

2

1 2

3 4

3

1 3 4

8 6 -2

-3 4 0

0

#### Sample Output

homogeneous

not homogeneous