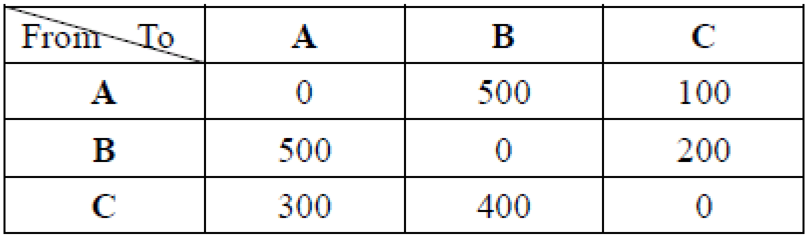
C Active Chemical Manipulation

☢*-- handle with care………*

A chemical factory is capable of producing a wide range of chemicals with economical value. Due to market demand, the factory may have to convert from one chemical to another type in order not to hold too much stock. However, the major problem is that a huge amount of heat will be generated whenever chemicals are converted. To maintain occupational safety, instead of converting from chemical A to chemical B directly, sometimes, the factory may convert chemical A to an intermediate chemical X first, and then from chemical X to finally chemical B (maybe through some more intermediates as well). For example:

From To	A	B	C
A	0	1000	100
B	600	0	200
C	300	400	0

To convert from **A** to **B** directly, 1000 units of heat will be generated. A better method would be changing from **A** to **C** first (100 units), and then from **C** to **B** (400 units), making it a total of 500 units.
Your task here is to calculate the *Active Chemical Manipulation Index* (*ACM Index*) of each chemical using the method mentioned below:
First of all, the program has to calculate the **minimum** amount of heat generated from one chemical to another through some *safer* path. *(You may go though unlimited number of intermediates or you may select direct conversion, if it generates less heat)*.
For example, the minima matrix for chemical **A**, **B**, and **C** mentioned above is:



From To	A	B	C
A	0	500	100
B	500	0	200
C	300	400	0

The *ACM Index* of a chemical is the median (middle value) of heat generated from that chemical to all others (including itself). For example, for chemical **A** (1st row), the median of {0,100,500} is 100. So the *ACM Index* of chemical **A** is 100.

Input

The input consist of multiple test cases, each case consist of an integer N (0<N<1000), representing the number of chemicals. The number is followed by a matrix, of N x N non-negative integers (smaller than 1000) showing the amount heat generated.
Numbers are separated by at least 1 space and there's no empty line between cases.
The input is terminated by EOF.

Output

The program has to output **N** numbers for each case, showing the *ACM Index* of each chemical. Each number occupies a single line with no leading/trailing spaces. **No fractional places should be printed if the *ACM Index* is integer**. (see sample output)
There should not be any empty lines between/after cases

Sample Input

```
3
0 1000 100
600 0 200
300 400 0
2
0 1
1 0
```

Output for Sample Input

```
100
200
300
0.5
0.5
```