

Not submitted yet

English (en)

Bessie has an $N \times N$ ($1 \leq N \leq 1000$) addition table where the integer in the cell at row r and column c is $r + c$, for all $1 \leq r, c \leq N$. For example, for $N = 3$, the table would look like this:

```
2 3 4
3 4 5
4 5 6
```

Unfortunately, Elsie got ahold of the table and permuted it by performing the following three types of operations as many times as she wanted.

1. Swap two rows
2. Swap two columns
3. Select two values a and b that are both present in the table, then simultaneously change every occurrence of a to b and every occurrence of b to a .

Elsie will always perform operations in increasing order of type; that is, she performs as many operations as she likes (possibly none) first of type 1, then of type 2, and finally of type 3.

Help Bessie recover a possible state of the table after Elsie finished applying all of her operations of types 1 and 2, but before applying any operations of type 3. There may be multiple possible answers, in which case you should output the lexicographically smallest one.

To compare two tables lexicographically, compare the first entries at which they differ, when reading both tables in the natural order (rows from top to bottom, left to right within a row).

INPUT FORMAT (input arrives from the terminal / stdin):

The first line contains N .

The next N lines each contain N integers, representing Bessie's addition table after Elsie has permuted it.

OUTPUT FORMAT (print output to the terminal / stdout):

The lexicographically minimum possible state of the table after all operations of types 1 and 2, but before any operations of type 3. It is guaranteed that the answer exists.

SAMPLE INPUT:

```
1
2
```

SAMPLE OUTPUT:

```
2
```

Regardless of what operations Elsie performs, the table won't change.

SAMPLE INPUT:

```
3
3 4 2
5 2 3
6 3 5
```

SAMPLE OUTPUT:

```
4 2 3
5 3 4
6 4 5
```

Here is a possible sequence of operations Elsie might have performed.

```
2 3 4
3 4 5
4 5 6
-> (op 1: swap columns 2 and 3)
2 4 3
3 5 4
4 6 5
-> (op 1: swap columns 1 and 2)
4 2 3
5 3 4
6 4 5
-> (op 3: swap values 2 and 3)
4 3 2
5 2 4
6 4 5
-> (op 3: swap values 3 and 4)
3 4 2
5 2 3
6 3 5
```

Note: the following is also a possible state of the table after operations of types 1 and 2, but it is not the lexicographically smallest because the second entry of the first row is larger than in the correct answer.

```
4 6 5
3 5 4
2 4 3
```

SAMPLE INPUT:

```
6
8 10 5 6 7 4
12 11 10 4 8 2
5 4 6 7 9 8
10 2 4 8 5 12
6 8 7 9 3 5
4 12 8 5 6 10
```

SAMPLE OUTPUT:

```
7 5 8 9 10 6
4 2 5 6 7 3
8 6 9 10 11 7
5 3 6 7 8 4
9 7 10 11 12 8
6 4 7 8 9 5
```

SCORING:

- Inputs 4-5: $N \leq 6$
- Inputs 6-7: $N \leq 8$
- Inputs 8-11: $N \leq 100$
- Inputs 12-15: No additional constraints.