[I2A] - Grade Sorting - Simplified

Time: 2 sec / Memory: 256 MB

Background

This is a simplified version of the original Grade Sorting (/mvZ7zxR6TiKK4eSKs8UB9g) problem.

Problem Statement

NCYU organized an entrance examination with five subjects:

> "Chinese",
> "English",
> "Mathematics",
> "Science", and
> "Social Studies".

The score of each subject is an integer between 0 and 100.

To select outstanding candidates, for any two students, the following rules are applied to determine the relative rank of them.

1. Exceptional Performance Priority:
   If one has scores of 90 or above in all five subjects and the other doesn't, then the one that does is ranked higher for admission.

2. Total Score Priority:
   If rule 1 does not provide an ordering for them, then the student with the higher total score across all five subjects has a higher priority for admission.

3. Subject-wise Priority:
   If the above rules are insufficient to determine the ranking between the two students, then compare their scores sequentially in the following order:

Chinese, English, Mathematics, Science, and Social Studies.
The student with the higher score in the first differing subject is given priority for admission.

This year, a total of n students are participating in the entrance examination. You are provided with each student's scores in all five subjects. Write a program to determine the priority ranking of each student based on the above rules.

Input

n

Chinese_1 English_1 Mathematics_1 Science_1 SocialStudies_1

Chinese_2 English_2 Mathematics_2 Science_2 SocialStudies_2

...

Chinese_n English_n Mathematics_n Science_n SocialStudies_n

Output

Print the scores of the students in the order determined by the above ranking rules. Each student's scores should be printed in a separate line.

Constraints

$1 \leq n \leq 10^5$
$0 \leq Chinese_i, English_i, Mathematics_i, Science_i, SocialStudies_i \leq 100$

It is guaranteed that for any two students, there is at least one subject in which their scores differ.

Example

Input:

```
4
91 100 100 97 89
90 91 90 90 90
92 100 100 96 89
90 90 92 90 90
```

Output:

```
90 90 92 90 90
90 91 90 90 90
92 100 100 96 89
91 100 100 97 89
```