Problem A: Aliases

Time limit: 40s, memory limit: 1GB.

The Individual Fast Programming Contest (IFPC) has seen a record number of registrations this year – n competitors. Your task is to set up accounts for them. Each account will have an *alias*, i.e. a username. The IFPC system only accepts aliases composed of lowercase English letters and numbers.

In order not to mess things up, you create aliases based on the first and last names of the participants, following the same algorithm: the first a letters of the first name, then the first b letters of the last name, and finally c digits of your choice. If the first name has fewer than a letters or the last name has fewer than b letters, you just take all the letters (then the alias is shorter, but it is fine). For example, for $a = b = c = 3$ James Bond could get the alias `jambon007` and Lady Di could get `laddi123`.

You want the aliases to be as short as possible – that saves a few bytes of memory and you like saving memory. On the other hand, all the participants must have unique aliases. For a given list of participants (their names and surnames), find a, b and c that will allow you to create different aliases for all participants, and for which $a+b+c$ will be as small as possible. Numbers a, b and c can equal 0 (but not all at the same time – the aliases must be non-empty).

Input

The first line of input contains the number of test cases z ($1 \leqslant z \leqslant 6$). The descriptions of the test cases follow.

The first line of the test case contains one integer n ($1 \leqslant n \leqslant 200\,000$) – the number of participants. In the following n lines there are the first and last names of the contestants. Both first and last name are sequences consisting only of lowercase letters of the English alphabet.

The sum of the lengths of all names and surnames of the competitors in one test case does not exceed 1 500 000. It is possible that two participants have the same name and surname (and they still need to get different aliases).

Output

For each test case, output three non-negative integers a, b and c that allow you to create unique aliases and have the lowest possible sum.

If there are several possible solutions with the minimal sum, you can output any of them.

Example

For an example input:	A possible correct output is:
1 11 sven eriksson erik svensson sven svensson erik eriksson bjorn eriksson bjorn svensson bjorn bjornsson erik bjornsson sven bjornsson thor odinsson odin thorsson	1 1 0

Explanation

Unique aliases can be obtained by taking the initials of all participants (the first letters of the first name and surname: `se`, `es`, `ss`, `ee`, ...). The answer `1 0 1` is also correct as you can create unique aliases by adding one number to the first letter of each person's name.