Messy Phone List

Adam has a lot of friends and therefore he stored a lot of phone numbers in his phone database. As his telephone doesn't belong to the latest generation, its database is somewhat simple. In fact, all entries are stored line by line, exactly in the way Adam once typed them in, no matter what format he used. Unfortunately he changed the format from time to time, unable or too lazy to remember how he did it the last time. So, after years have passed (Adam likes his phone and doesn't want to replace it by a modern one), his phone number list has become really messed up and he wants to do some clean up. Your task is to write a program that will do that clean up for him.

Every entry consists of three parts: telefon number, first name and last name of one of Adam's friends. The order of these parts may vary. An entry may start with the phone number, following the name or vice versa, phone number and name always separated by exactly one space. The order of the two parts of the name may also vary. Either it is first name before last name, separated by a space, or it is last name before first name, separated by a comma.
A phone number may contain an optional leading area code, separated by a "-" or a "/" from the local code. If an area code is missing, the area code of Adam's hometown is assumed. Area code and local code both consist of at least three and at most ten digits. There may be additional optional spaces in between for better readability. For the final list the phone numbers have to be normalized. A normalized phone number consists of the area code and the local code, separated by "-" without any spaces.
A name consists of any letters of the English alphabet. Capital letters and small letters may be mixed, as Adam didn't pay attention to that when he typed in the entries. First name and last name each are at least one and at most 20 characters long. For the final list all names have to be normalized. A normalized name consists of only small letters, except the first one being a capital letter. All following explanations refer to normalized numbers and names.

If two different persons have the same phone number, that will be considered as being an error - none of these entries must appear in the final list. If two entries are equal, they must appear only once in the final list. If a person is listed with different phone numbers, this person has changed phone numbers over the years and will only appear with the latest of the listed numbers, that does not have to be removed because of the reason stated above.

Input

Input starts with a positive integer t (t<50) in a single line, then t testcases follow. Every testcase starts with one line containing a positive integer n (n<1000), the number of phone list entries, and - separated by a space - the area code of Adam's hometown. Then n lines follow, each line representing one entry in the phone list.

Output

For each testcase first print the number k of entries in the final list in a single line. Then k lines must follow, the cleaned up phone list. Each line is a single entry that has to look exactly like this: [first name] [last name]: [phone number]. Square brackets only for clarity. The list has to be sorted in alphabetical order according to last names (primary key) and - if necessary - first names (secondary key).

Example

Input:

```
2
10 0608
Sastre,Carlos 030/64 736 666
Voigt,Jens 07401-4498
A Winokurow 0289-334405
Jan ULLRICH 089-77 98 00 9
089/779 8009 Ullrich,Jan
LANCE Armstrong 0608 / 220 4 768 86
Jan Ullrich 089/7798 005
02 89 / 33 44 05 Contador,A
ArmStrong,Lance 220476886
Ullrich,JaN 0289 / 334405
5 012
Becker,Franz 1200344
Becker,Boris 034/50005
Boris Becker 012 / 50 005
5000 5 Boris Becker
Franz Beckenbauer 332323
```

Output:

```
4
Lance Armstrong: 0608-220476886
Carlos Sastre: 030-64736666
Jan Ullrich: 089-7798005
Jens Voigt: 07401-4498
3
Franz Beckenbauer: 012-332323
Boris Becker: 012-50005
Franz Becker: 012-1200344
```