One of the great things about movies on DVD is that you can choose the language for the subtitles, in case you need them.

Let us consider that in the DVD the subtitles for a given language come in a text file, one subtitle in each line. By "one subtitle" we mean the short phrase that we see on the screen at any given time, even if that phrase uses more than one line on the screen. The subtitle files for all the available languages have the same number of lines, although some lines can be empty, for lack of an appropriate translation. There will be an extra file indicating the exact times in which each subtitle should be displayed on the screen, but we shall ignore it in this problem.

Today, we want to explore the subtitle files in order to automatically construct simple bilingual dictionaries. The idea is quite simple. Take, for example, Portuguese and English subtitles. Suppose a given word p appears in the Portuguese file in lines x_1, x_2, ..., x_n, with $n \geq 2$. Now suppose a word e appears in exactly the same lines in the English file. Then we can safely assume e is a legitimate translation of p, and vice-versa. Here is an example. First, a text in Portuguese:

```
Por favor, mostre-me o seu bilhete.
Lamento, mas esqueci-me dele.
Esqueceu-se do bilhete? Voce deve ser maluco.
Eu nao sou maluco. A sua mae e que e.
```

Note that we are not using diacritical marks in the Portuguese text. Now, the equivalent text in English:

```
May I see your ticket, please?
I am sorry, but I forgot it.
So, you forgot the ticket? You must be crazy.
I am not crazy. Crazy is your mother.
```

The words `bilhete` and `ticket` appear in lines 1 and 3, in the Portuguese and English texts, respectively. We, therefore, conclude that `bilhete` is the Portuguese translation of `ticket`. Likewise for `maluco` and `crazy` in lines 3 and 4, even though `crazy` appears twice in line 4 and `maluco` only appears once. The word `forgot` appears in lines 2 and 3, but cannot be translated using this method, as in the Portuguese text the corresponding words in these lines, `esqueci-me` and `esqueceu-se`, are different.

More generally, we can `say` that a word w_1 in language L_1 translates to word w_2 in language L_2 if the set of line numbers in which w_1 occurs is equal to the set of line numbers in which w_2 appears in their respective subtitle files. We consider only words that occur at least twice, and words with at least three characters. Note that several occurrences of the same word on a given line are equivalent to a single occurrence.

There's an ambiguous case, however: two (or more) words in the same language appearing on exactly the same lines in the subtitle file. To remove this ambiguity, those words are to be appended, separated by one space, in alphabetical order. See the example below, in the input and output samples.

The result of processing is a sorted list of pairs of words, as in a dictionary. The list is be sorted by the first word of each pair and all words are in lowercase only.

Your task is to write a program that given two subtitle files constructs a bilingual dictionary as explained above. To avoid difficult cases caused by small words, your program should ignore all input words with less than three characters.

Input

The first line of the input contains the number T of test cases, followed by T input blocks.

The first line of each input block contains the number of lines in each language, $0 < N \leq 1000$. The following N lines contain the subtitles in the first language. After that, N more lines contain the subtitles of the second language.

Each subtitle consists of normal ASCII text, using both lowercase and uppercase, and punctuation, but no diacritical marks (accents, cedillas or tildes). For the purposes of this program, words contain letters and hyphens only, and processing is case insensitive.

You can assume that the maximum number of words in each subtitle line is 20; the maximum number of different words for each language is 1000; and the maximum length of each word is 24.

Output

The output for each test case is a sorted list of lines, each one containing two lowercase words separated by a slash. The second word in each pair is the translation of the first one, as discovered by the program. In some cases, there might be pairs or triples, etc., of words, as a result of the ambiguity referred above.

Separate the output of each test case with a single blank line.

Sample Input

```
1
10
Quero um copo de cerveja, bem fresca.
Nao temos cerveja, mas temos vinho.
Nao obrigado, vinho nao quero.
Tambem temos sumo de laranja natural.
Esta bem, entao quero um sumo de laranja.
Mais alguma coisa?
Sim, um pastel de nata.
Com certeza. Um sumo e um pastel. Sao quatro euros.
Quatro euros!!! Mas isso e um roubo.
Se acha que e um roubo, chame a policia.
I want a glass of beer, very cool.
We dont have beer, but we have wine.
No thanks, I dont want wine.
We also have natural orange juice.
OK, then I want one orange juice.
Anything else?
Yes, a cream cake.
Of course. One juice and one cake. Thats four euros.
Four euros!!! That is a theft.
If you think it is a theft, call the police.
```

Sample Output

```
cerveja/beer
euros quatro/euros four
laranja/orange
nao vinho/dont wine
pastel/cake
quero/want
roubo/theft
sumo/juice
temos/have
```