Problem 10420

Problem B
List of Conquests
Input: standard input
Output: standard output
Time Limit: 2 seconds

In Act I, Leporello is telling Donna Elvira about his master's long list of conquests:

> ``This is the list of the beauties my master has loved, a list I've made out myself: take a look, read it with me. In Italy six hundred and forty, in Germany two hundred and thirty-one, a hundred in France, ninety-one in Turkey; but in Spain already a thousand and three! Among them are country girls, waiting-maids, city beauties; there are countesses, baronesses, marchionesses, princesses: women of every rank, of every size, of every age.'' (*Madamina, il catalogo è questo*)

As Leporello records all the ``beauties'' Don Giovanni ``loved'' in chronological order, it is very troublesome for him to present his master's conquest to others because he needs to count the number of ``beauties'' by their nationality each time. You are to help Leporello to count.

Input

The input consists of at most **2000** lines, but the first. The first line contains a number **n**, indicating that there will be **n** more lines. Each following line, with at most **75** characters, contains a country (the first word) and the name of a woman (the rest of the words in the line) Giovanni loved. You may assume that the name of all countries consist of only one word.

Output

The output consists of lines in alphabetical order. Each line starts with the name of a country, followed by the total number of women Giovanni loved in that country, separated by a space.

Sample Input

```
3
Spain Donna Elvira
England Jane Doe
Spain Donna Anna
```

Sample Output

```
England 1
Spain 2
```

Problem 299

Train Swapping

At an old railway station, you may still encounter one of the last remaining ``train swappers''. A train swapper is an employee of the railroad, whose sole job it is to rearrange the carriages of trains.

Once the carriages are arranged in the optimal order, all the train driver has to do, is drop the carriages off, one by one, at the stations for which the load is meant.

The title ``train swapper'' stems from the first person who performed this task, at a station close to a railway bridge. Instead of opening up vertically, the bridge rotated around a pillar in the center of the river. After rotating the bridge 90 degrees, boats could pass left or right.

The first train swapper had discovered that the bridge could be operated with at most two carriages on it. By rotating the bridge 180 degrees, the carriages switched place, allowing him to rearrange the carriages (as a side effect, the carriages then faced the opposite direction, but train carriages can move either way, so who cares).

Now that almost all train swappers have died out, the railway company would like to automate their operation. Part of the program to be developed, is a routine which decides for a given train the least number of swaps of two adjacent carriages necessary to order the train. Your assignment is to create that routine.

Input Specification

The input contains on the first line the number of test cases (*N*). Each test case consists of two input lines. The first line of a test case contains an integer *L*, determining the length of the train (0<=L<=50). The second line of a test case contains a permutation of the numbers 1 through *L*, indicating the current order of the carriages. The carriages should be ordered such that carriage 1 comes first, then 2, etc. with carriage *L* coming last.

Output Specification

For each test case output the sentence: '`Optimal train swapping takes` *S* `swaps.`' where *S* is an integer.

Example Input

```
3
3
1 3 2
4
4 3 2 1
2
2 1
```

Example Output

```
Optimal train swapping takes 1 swaps.
Optimal train swapping takes 6 swaps.
Optimal train swapping takes 1 swaps.
```

Problem 340

Master-Mind Hints

MasterMind is a game for two players. One of them, *Designer*, selects a secret code. The other, *Breaker*, tries to break it. A code is no more than a row of colored dots. At the beginning of a game, the players agree upon the length *N* that a code must have and upon the colors that may occur in a code.

In order to break the code, Breaker makes a number of guesses, each guess itself being a code. After each guess Designer gives a hint, stating to what extent the guess matches his secret code.

In this problem you will be given a secret code s1...sn and a guess g1...gn , and are to determine the hint. A hint consists of a pair of numbers determined as follows.

A *match* is a pair (*i*,*j*), 1<=i<=n and 1<=j<=n , such that si=gi . Match (*i*,*j*) is called *strong* when *i* = *j*, and is called *weak* otherwise. Two matches (*i*,*j*) and (*p*,*q*) are called *independent* when *i* = *p* if and only if *j* = *q*. A set of matches is called *independent* when all of its members are pairwise independent.

Designer chooses an independent set *M* of matches for which the total number of matches and the number of strong matches are both maximal. The hint then consists of the number of strong followed by the number of weak matches in *M*. Note that these numbers are uniquely determined by the secret code and the guess. If the hint turns out to be (*n*,0), then the guess is identical to the secret code.

Input

The input will consist of data for a number of games. The input for each game begins with an integer specifying *N* (the length of the code). Following these will be the secret code, represented as *N* integers, which we will limit to the range 1 to 9. There will then follow an arbitrary number of guesses, each also represented as *N* integers, each in the range 1 to 9. Following the last guess in each game will be *N* zeroes; these zeroes are not to be considered as a guess.

Following the data for the first game will appear data for the second game (if any) beginning with a new value for *N*. The last game in the input will be followed by a single zero (when a value for *N* would normally be specified). The maximum value for *N* will be 1000.

Output

The output for each game should list the hints that would be generated for each guess, in order, one hint per line. Each hint should be represented as a pair of integers enclosed in parentheses and separated by a comma. The entire list of hints for each game should be prefixed by a heading indicating the game number; games are numbered sequentially starting with 1. Look at the samples below for the *exact* format.

Sample Input

```
4
1 3 5 5
1 1 2 3
4 3 3 5
6 5 5 1
6 1 3 5
1 3 5 5
0 0 0 0
10
1 2 2 2 4 5 6 6 6 9
1 2 3 4 5 6 7 8 9 1
1 1 2 2 3 3 4 4 5 5
1 2 1 3 1 5 1 6 1 9
1 2 2 5 5 5 6 6 6 7
0 0 0 0 0 0 0 0 0 0
0
```

Sample Output

```
Game 1:
    (1,1)
    (2,0)
    (1,2)
    (1,2)
    (4,0)
Game 2:
    (2,4)
    (3,2)
    (5,0)
    (7,0)
```