Problem 2.A Optimal Parking

When shopping on Long Street, Michael usually parks his car at some random location, and then walks to the stores he needs. Can you help Michael choose a place to park which minimises the distance he needs to walk on his shopping round?

Long Street is a straight line, where all positions are integer. You pay for parking in a specific slot, which is an integer position on Long Street. Michael does not want to pay for more than one parking though. He is very strong, and does not mind carrying all the bags around.

Input

The first line of input gives the number of test cases, $1 \leq t \leq 100$. There are two lines for each test case. The first gives the number of stores Michael wants to visit, $1 \leq n \leq 20$, and the second gives their n integer positions on Long Street, $0 \leq x_i \leq 99$.

Output

Output for each test case a line with the minimal distance Michael must walk given optimal parking.

Sample Input

```
2
4
24 13 89 37
6
7 30 41 14 39 42
```

Sample Output

```
152
70
```

Problem 2.B Secret Research

At a certain laboratory results of *secret research* are thoroughly encrypted. A result of a single experiment is stored as an information of its completion:

'positive result', 'negative result', 'experiment failed' or 'experiment not completed'

The encrypted result constitutes a string of digits S, which may take one of the following forms:

- positive result — $S = 1$ or $S = 4$ or $S = 78$
- negative result — $S = S35$
- experiment failed — $S = 9S4$
- experiment not completed — $S = 190S$

(A sample result $S35$ means that if we add digits 35 from the right hand side to a digit sequence then we shall get the digit sequence corresponding to a failed experiment)

You are to write a program which decrypts given sequences of digits.

Input

A integer n stating the number of encrypted results and then consecutive n lines, each containing a sequence of digits given as ASCII strings.

Output

For each analysed sequence of digits the following lines should be sent to output (in separate lines):

+	for a positive result
-	for a negative result
*	for a failed experiment
?	for a not completed experiment

In case the analysed string does not determine the experiment result, a first match from the above list should be outputted.

Sample Input

```
4
78
7835
19078
944
```

Sample Output

```
+
-
?
*
```

Problem 2.C SMS Typing

Cell phones have become an essential part of modern life. In addition to making voice calls, cell phones can be used to send text messages, which are known as SMS for short. Unlike computer keyboards, most cell phones have limited number of keys. To accommodate all alphabets, letters are compacted into single key. Therefore, to type certain characters, a key must be repeatedly pressed until that character is shown on the display panel.

In this problem we are interested in finding out the number of times keys on a cell phone must be pressed to type a particular message.

In this problem we will assume that the key pad of our cell phone is arranged as follows.

```
---------------------
|      | abc | def  |
---------------------
| ghi  | jkl | mno  |
---------------------
| pqrs | tuv | wxyz |
---------------------
|      | <SP>|      |
---------------------
```

In the above grid each cell represents one key. Here `<SP>` means a space. In order to type the letter '`a`', we must press that key once, however to type '`b`' the same key must be repeatedly pressed twice and for '`c`' three times. In the same manner, one key press for '`d`', two for '`e`' and three for '`f`'. This is also applicable for the remaining keys and letters. Note that it takes a single press to type a space.

Input

The first line of input will be a positive integer T where T denotes the number of test cases. T lines will then follow each containing only spaces and lower case letters. Each line will contain at least 1 and at most 100 characters.

Output

For every case of input there will be one line of output. It will first contain the case number followed by the number of key presses required to type the message of that case. Look at the sample output for exact formatting.

Sample Input

```
2
welcome to ulab
good luck and have fun
```

Sample Output

```
Case #1: 29
Case #2: 41
```

Problem 2.D Counterfeit Dollar

Sally Jones has a dozen Voyageur silver dollars. However, only eleven of the coins are true silver dollars; one coin is counterfeit even though its color and size make it indistinguishable from the real silver dollars. The counterfeit coin has a different weight from the other coins but Sally does not know if it is heavier or lighter than the real coins.

Happily, Sally has a friend who loans her a very accurate balance scale. The friend will permit Sally three weighings to find the counterfeit coin. For instance, if Sally weighs two coins against each other and the scales balance then she knows these two coins are true. Now if Sally weighs one of the true coins against a third coin and the scales do not balance then Sally knows the third coin is counterfeit and she can tell whether it is light or heavy depending on whether the balance on which it is placed goes up or down, respectively.

By choosing her weighings carefully, Sally is able to ensure that she will find the counterfeit coin with exactly three weighings.

Input

The first line of input is an integer n ($n > 0$) specifying the number of cases to follow. Each case consists of three lines of input, one for each weighing. Sally has identified each of the coins with the letters `A`–`L`. Information on a weighing will be given by two strings of letters and then one of the words "`up`", "`down`", or "`even`". The first string of letters will represent the coins on the left balance; the second string, the coins on the right balance. (Sally will always place the same number of coins on the right balance as on the left balance.) The word in the third position will tell whether the right side of the balance goes up, down, or remains even.

Output

For each case, the output will identify the counterfeit coin by its letter and tell whether it is heavy or light. The solution will always be uniquely determined.

Sample Input

```
1
ABCD EFGH even
ABCI EFJK up
ABIJ EFGH even
```

Sample Output

```
K is the counterfeit coin and it is light.
```

Problem 2.E List of Conquests

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 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. 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
```