

Lab 11 Tasks

Q1. Cryptanalysis is the process of breaking someone else's cryptographic writing. This sometimes involves some kind of statistical analysis of a passage of (encrypted) text. Your task is to write a program which performs a simple analysis of a given text.

Input

The first line of input contains a single positive decimal integer n . This is the number of lines which follow in the input. The next n lines will contain zero or more characters (possibly including whitespace). This is the text which must be analyzed.

Output

Each line of output contains a single uppercase letter, followed by a single space, then followed by a positive decimal integer. The integer indicates how many times the corresponding letter appears in the input text. Upper and lower case letters in the input are to be considered the same. No other characters must be counted. The output must be sorted in descending count order; that is, the most frequent letter is on the first output line, and the last line of output indicates the least frequent letter. If two letters have the same frequency, then the letter which comes first in the alphabet must appear first in the output. If a letter does not appear in the text, then that letter must not appear in the output.

Sample Input	Sample Output
3	S 7
This is a test.	T 6
Count me 1 2 3 4 5.	I 5
Wow!!!! Is this question easy?	E 4
	O 3
	A 2
	H 2
	N 2
	U 2
	W 2
	C 1
	M 1
	Q 1
	Y 1

Q2. Your younger brother has just finished his homework for the part “additions and subtractions for integers not greater than one hundred” and asks you to check the answers. Each question (together with the answer computed by your younger brother) is formatted either as $a + b = c$ or $a - b = c$, where a and b are numbers prepared by the teacher (they are guaranteed to be non-negative integers not greater than 100), c is the answer computed by your younger brother and is either a nonnegative integer not greater than 200, or a single character ‘?’ (that means, he is unable to compute the answer).

Input

The first line will specify how many inputs there will be. There will be at most 100 lines in the input. Each line contains a question with your younger brother's answer, formatted as stated above. There will be no space characters in each line (excluding the newline character). Numbers will never have leading zeros.

Output

Print a single integer in a line, the number of correct answers.

Sample Input	Sample Output
4	2
1+2=3	
3-1=5	
6+7=?	
99-0=99	

Q3. Write a program in C to demonstrate how to handle the pointers in the program. The expected output is given below:

```
Address of m : 0x7ffcc3ad291c
Value of m : 29

Now ab is assigned with the address of m.
Address of pointer ab : 0x7ffcc3ad291c
Content of pointer ab : 29

The value of m assigned to 34 now.
Address of pointer ab : 0x7ffcc3ad291c
Content of pointer ab : 34

The pointer variable ab is assigned with the value 7 now.
Address of m : 0x7ffcc3ad291c
Value of m : 7
```

Here *m* is an integer variable and *ab* is a pointer. Follow the same format as above when printing in the console.

Q4. Write a program in C to demonstrate the use of &(address of) and *(value at address) operator. The expected output is given below:

```
Using & operator :
-----
address of m = 0x7ffea3610bb8
address of fx = 0x7ffea3610bbc
address of cht = 0x7ffea3610bb7

Using & and * operator :
-----
value at address of m = 300
value at address of fx = 300.600006
value at address of cht = z

Using only pointer variable :
-----
address of m = 0x7ffea3610bb8
address of fx = 0x7ffea3610bbc
address of cht = 0x7ffea3610bb7

Using only pointer operator :
-----
value at address of m = 300
value at address of fx= 300.600006
value at address of cht= z
```

Here *m* is an integer, *fx* is a float and *cht* is a character type variable. The two sections will be printed by directly using these variables. Use *& together for the second section. The last two sections will be printed by using pointers to the variables. Follow the same format as above when printing in the console

Q5. Given any positive integer, if we permute its digits, the difference between the numbers we get and the given number will always be divisible by 9. For example, if the given number is 123, we may rearrange the digits to get 321. The difference = 321 - 123 = 198, which is a multiple of 9 (198 = 9 × 22).

Input and Output

Each line of input gives a positive integer *n* (≤ 2000000000). You are to find two integers *a* and *b* formed by rearranging the digits of *n*, such that *a-b* is maximum. *a* and *b* should NOT have leading zeros. You should then show that *a-b* is a multiple of 9, by expressing it as 9 × *k*, where *k* is an integer.

Sample Input	Sample Output
123	321 - 123 = 198 = 9 * 22
2468	8642 - 2468 = 6174 = 9 * 686
1000	1000 - 1000 = 0 = 9 * 0
10101	11100 - 10011 = 1089 = 9 * 121