



**PRO
SOL
VE**

8.0

TALK IS CHEAP, SHOWS US THE CODE

Competitive Programming

MOCK

This problem set contains 5 questions (A-E)

23 November 2019

Organized by

Artificial Intelligence Society (AIS) UiTM Shah Alam

A. Spaced Out

Given 3 different integer numbers, determine if the difference between the smallest number and the middle number is the same as the difference between the middle number and the largest number. The numbers can be entered in any order. If the differences are exactly the same the program should indicate the numbers are "Spaced Out" otherwise the numbers are "NOT Spaced Out".

Input

The first line of input will be an n integer, which represent the number of cases, followed by three (3) distinct integers (separated by line) for each line. The input should be in range of 1 until 10.

Output

The output should be either 'Spaced Out' for the same in differences and 'NOT Spaced Out' for otherwise. Be sure there are no extra spaces in front and back of the word.

Sample Input / Output

Input	Output
3	
3 5 7	Spaced Out
7 5 3	Spaced Out
5 9 6	NOT Spaced Out

B. Look-and-Say Sequence

Given a number, display the next number in the look-and-say sequence. The look-and-say sequence is determined by looking at the number, saying the number of times that each digit appears consecutively and then the digit itself. For example, given 3111 the next number in sequence would be 1331 because the number entered (3111) was composed of one 3 followed by three 1s.

Input

The first line of input should be the number of test cases (n), followed by the numbers for n lines. The numbers must be more than zero (refer to positive numbers).

Output

The output should be look-and-say sequence as described above. Refer Sample Input / Output table below;

Sample Input / Output

Input	Output
3	
3111	1331
1211	111221
111221	312211

C. Collatz Conjecture

The Collatz Conjecture (also known as the Ulam conjecture or the Syracuse problem) is an unsolved mathematical problem, which is very easy to formulate:

1. Take any natural number
2. If the value is even, half it, otherwise triple it and add one
3. Repeat step 2, until you reach 4, 2, 1 sequence
4. You will ALWAYS reach 1, eventually.

For Example:

$x = 17$

- | | | |
|------|-------------------|---|
| (1) | $17 * 3 + 1 = 52$ | |
| (2) | $52 / 2 = 26$ | |
| (3) | $26 / 2 = 13$ | |
| (4) | $13 * 3 + 1 = 40$ | |
| (5) | $40 / 2 = 20$ | |
| (6) | $20 / 2 = 10$ | |
| (7) | $10 / 2 = 5$ | |
| (8) | $5 * 3 + 1 = 16$ | |
| (9) | $16 / 2 = 8$ | |
| (10) | $8 / 2 = 4$ | |
| (11) | $4 / 2 = 2$ | |
| (12) | $2 / 2 = 1$ | |
| (13) | $1 * 3 + 1 = 4$ | ← The first value of second repetition of 4, 2, 1 |

...

The last sequence: 4, 2, 1 is an infinitely repeating loop. The formulated conjecture is that for any of x , the sequence will always reach 4, 2, 1 ultimately.

While the problem cannot be proved, the assignment is to write a code and print out the number of iteration to reach the second 4 of 4, 2, 1 of repetition.

Input

The first line should be the number of test cases (n), followed by n lines of integer. The value should be in between 10 until 100.

Output

The output should be the number of iteration require to reach the first value of second repetition of 4, 2, 1. As above example, the output should be 13.

Sample Input / Output

Input	Output
3	
35	14
62	108
87	31

D. Word Score!

Word score is a simple game that calculate each letter in a word (which each letter has it own value) and compare those words to identify the winner (based on highest score). It sounds unfair to “treat” like that for words, but I guess its how life treating people, right? Different ‘level’ will receive different way of treatment from others. Hm.

...

Fine, don’t mind me. Let’s get back to the game!

The score of a word is calculated based on its character values (not case-sensitive), according to this table below;

Letters	Point
e, a, l, o, n, r, t, i, s, u	1
d, g	2
b, c, m, p	3
f, h, v, w, y	4
k	5
j, x	8
q, z	10

Strict Rule!

The maximum length of a word is 10 letters and must **ONLY** letters. No number and other symbol allowed. If exist, display ‘Disqualified’ as the output (even there is only one word that cause the problem).

Input

The first line should be the number of test cases, n , followed by 4 different words (separated by space) for n lines.

Output

The output should be the word that win the game (the highest score) followed by its score in the parenthesis after a space. For any input that contains number or special characters will display ‘Disqualified’.

Sample Input / Output

Input	Output
3	
JAVASCRIPT RUBY JAVA PYTHON	JAVASCRIPT (24)
Alphabet Character Number Symbol	Character (16)
A4 Paper Spoon White	Disqualified

E. Encrypt Caesar Cipher

The Caesar cipher is a type of substitution cipher in which each letter in the plaintext is **shifted** to a certain number of places down the **alphabet**. For example, with a shift of 1, A would be replaced by B, B would become C, and so on. The method is named after Julius Caesar, who allegedly used it to communicate with his generals.

For Example:

Input: "defend the east wall of the castle"

Output: "efgfoe uif fbtu xbm pg uif dbtumf"

Input

The first line should be the number of test cases, n , followed by a text string for each n lines.

Output

The output should be the encrypted string.

Sample Input / Output

Input	Output
2	
Hello World	Ifmmp!Xpsme
supercalifragilisticexpialidocious	tvqf sdmjgsbhjmjtujdfyqjbmjepdj pvt