Problem A : Return Path (blue balloon)

For security reasons, it is advised that the return path of the president car should not be the same as the starting path.

You have been asked to write a program that can tell if the return path is the same as the starting path.

Standard Input

The first line of input contains a single integer **P**, ($1 \leq$ **P** ≤ 1000), which is the number of data sets that follow. Each data set consists of two lines containing each, one path which is composed of strings of less than 200 characters. Each string is a name of town. There will not be more than 20 towns per path. The name of town will be case insensitive and will be a string formed with characters [a-z] [A-Z] and with the sign “-”

Standard Output

For each data set, generate one line of output containing a single word “Same” if the starting path is same as the return path or “Different” if not.

Sample Input	Sample Output
3 Tsevie Notse Notse tsevie Lome Dapaong Kara Kara Lome Dapaong Aneho Kara Kpalime Sokode Sokode Kpalime kara aneho	Same Different Same

Problem B : Equation (red balloon)

You have been asked to write a program that can solve a linear equation.

Standard Input

The first line of input contains a single integer **P**, $(1 \leq \mathbf{P} \leq 1000)$, which is the number of data sets that follow. Each data set consists of two lines containing each, one member of the equation. All equations are composed of strings of less than 200 characters.

Each line of each equation will be in the form of **a*x*,** followed by a single space, followed by a sign “+”, followed by a single space, followed by **b*y***, followed by a single space, followed by a sign “=”, followed by a single space, followed by c.

$$\boxed{a\boldsymbol{x} + b\mathbf{y} = c}$$

where ***x*** and **y** are the variables (real number) and **a**, **b**, **c** are positive integers.

Standard Output

For each data set, generate two lines of output. The first line will contain “Equation **n**” where **n** is the number of the data set. The second line will contain the following answer:

- If the equation has no solution, print "No solution.".
- If the equation has infinitely many solutions, print "More than one solution.".
- If the equation has exactly one solution for (x,y), print "x = *solution1*", followed by two spaces, followed by "y = *solution2*" where *solution1 and solution2* are replaced by the appropriate reals numbers (printed to six decimals).

Print a blank line after each data set case output.

Sample Input	Sample Output
5	Equation 1
2x + 3y = 4	x = 2.000000 y = 0.000000
3x + 2y = 6	
124x + 20y = 160	Equation 2
2x + 4y = 5	x = 1.184210 y = 0.657894
123x + 321y = 123	
12x + 21y = 12	Equation 3
2x + 3y = 4	x = 1.000000 y = 0.000000
0x + 0y = 4	
123456x + 7y = 2000	Equation 4
123456x + 7y = 2000	No solution.
	Equation 5
	More than one solution.

Problem C : Cyclic Number (yellow balloon)

A cyclic number is an integer n digits in length which, when multiplied by any integer from 1 to n, yields a “cycle” of the digits of the original number. That is, if you consider the number after the last digit to “wrap around” back to the first digit, the sequence of digits in both numbers will be the same, though they may start at different positions.

For example, the number 142857 is cyclic, as illustrated by the following table:

142857 × 1 = 142857

142857 × 2 = 285714

142857 × 3 = 428571

142857 × 4 = 571428

142857 × 5 = 714285

142857 × 6 = 857142

Write a program which will determine whether or not numbers are cyclic.

Standard Input

The input will begin with a single integer **P** on the first line, indicating the number of cases that will follow.

The remaining lines of the input will consist of one integer per line, from 2 to 60 digits in length. (Note that preceding zeros should not be removed, they are considered part of the number and count in determining n. Thus, “01” is a two-digit number, distinct from “1” which is a one-digit number.)

Standard Output

For each input integer, write a line in the output indicating whether or not it is cyclic.

Sample Input	Sample Output
5 142857 142856 142858 01 0588235294117647	142857 is cyclic 142856 is not cyclic 142858 is not cyclic 01 is not cyclic 0588235294117647 is cyclic

Problem D : Minimun Steps (green balloon)

You are in a grid composed of huts. You can only move to another hut near your hut, with one step. One hut is considered near one another if the two huts have a side in common.

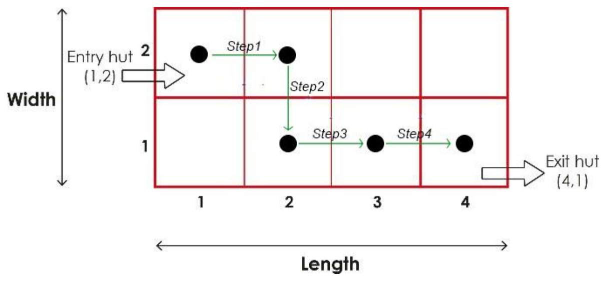
The grid can be seen as a rectangle of huts of width **W** and length **L**. **W** and **L** are integers.

A hut is identified by its position on the horizontal side and by its position on the vertical side of the grid.

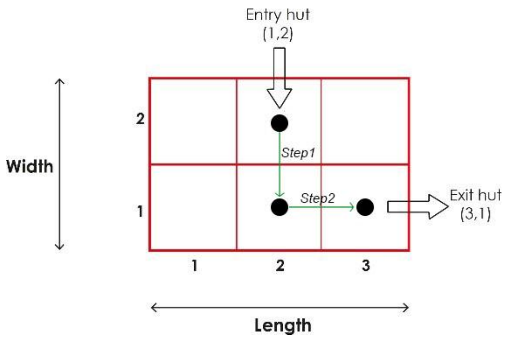
In this grid, all huts are "entry huts" but there is only one "exit hut".

Given the grid, your initial hut (called entry hut) and the exit hut, your task is to determine the minimum number of steps to go from the entry hut to the exit hut.

In the example of Grid A (grid of length 4 and width 2), the minimum number of steps is 4. In the second example (grid of length 3 and width 2), minimum number of steps is 2.



Grid A



Grid B

Standard Input

The input will begin with a single integer **P** on the first line, indicating the number of cases that will follow.

Each case consists of a single line made of 6 natural numbers with the following format:

L W A B C D where :

- **L** is the length of the grid and **W** is the width of the grid ;
- **A** is the length of the entry hut and **B** is the width of the entry hut ;
- **C** is the length of the exit hut and **D** is the width of the exit hut.

Standard Output

For each grid, print the minimum number of steps.

Sample Input	Sample Output

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

Problem E : Excel (white balloon)

A certain spreadsheet program labels the columns of a spreadsheet using letters. Column 1 is labeled as "A", column 2 as "B", …, column 26 as "Z". When the number of columns is greater than 26, another letter is used. For example, column 27 is "AA", column 28 is "AB" and column 52 is "AZ". It follows that column 53 would be "BA" and so on. Similarly, when column "ZZ" is reached, the next column would be "AAA", then "AAB" and so on.

The rows in the spreadsheet are labeled using the row number. Rows start at 1.

The designation for a particular cell within the spreadsheet is created by combining the column label with the row label. For example, the upper-left most cell would be "A1". The cell at column 55 row 23 would be "BC23".

You will write a program that converts numeric row and column values into the spreadsheet designation.

Standard Input

Input consists of lines of the form: **R*n*C*m***. ***n*** represents the row number [1,300000000] and ***m*** represents the column number, **1 <= *m* <= 300000000**. The values ***n*** and ***m*** define a single cell on the spreadsheet. Input terminates with the line: **R0C0** (that is, ***n*** and ***m*** are 0). There will be no leading zeroes or extra spaces in the input.

Standard Output

For each line of input (except the terminating line), you will print out the spreadsheet designation for the specified cell as described above.

Sample Input	Sample Output
R1C1 R3C1 R1C3 R299999999C26 R52C52 R53C17576 R53C17602 R0C0	A1 A3 C1 Z299999999 AZ52 YYZ53 YZZ53

Problem F : CheckSum (Orange balloon)

A checksum is an algorithm that scans a packet of data and returns a single number. The idea is that if the packet is changed, the checksum will also change, so checksums are often used for detecting transmission errors, validating document contents, and in many other situations where it is necessary to detect undesirable changes in data.

For this problem, you will implement a checksum algorithm called Quicksum. A Quicksum packet allows only uppercase letters and spaces. It always begins and ends with an uppercase letter. Otherwise, spaces and letters can occur in any combination, including consecutive spaces.

A Quicksum is the sum of the products of each character's position in the packet times the character's value. A space has a value of zero, while letters have a value equal to their position in the alphabet. So, A=1, B=2, etc., through Z=26. Here is an example Quicksum calculation for the packet "ACM" :

```
ACM: 1*1  + 2*3 + 3*13 = 46
```

Standard Input

The input consists of one or more packets followed by a line containing only # that signals the end of the input. Each packet is on a line by itself, does not begin or end with a space, and contains from 1 to 255 characters.

Standard Output

For each packet, output its Quicksum on a separate line in the output.

Sample Input	Sample Output
ACM MAPCOM NATIONAL PROGRAMMING CONTEST TOGO A C M ABC BBC #	46 228 4717 131 75 14 15