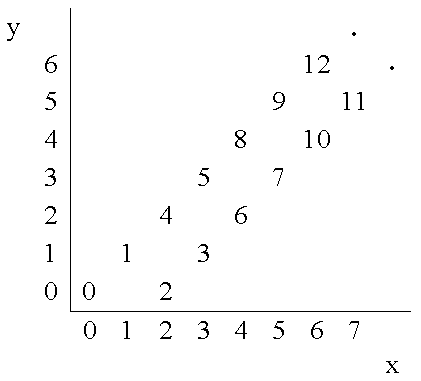
Starting from point (0,0) on a plane, we have written all non-negative integers 0, 1, 2,... as shown in the figure. For example, 1, 2, and 3 has been written at points (1,1), (2,0), and (3, 1) respectively and this pattern has continued.



You are to write a program that reads the coordinates of a point (x, y), and writes the number (if any) that has been written at that point. (x, y) coordinates in the input are in the range 0...10000.

**Input**

The first line of the input is N, the number of test cases for this problem. In each of the N following lines, there is x, and y representing the coordinates (x, y) of a point.

**Output**

For each point in the input, write the number written at that point or write No Number if there is none.

**Example**

**Input:**

3

4 2

6 6

3 4

**Output:**

6

12

No Number

Mo and Larry have devised a way of encrypting messages. They first decide secretly on the number of columns and write the message (letters only) down the columns, padding with extra random letters so as to make a rectangular array of letters. For example, if the message is “There’s no place like home on a snowy night” and there are five columns, Mo would write down

t o i o y

h p k n n

e l e a i

r a h s g

e c o n h

s e m o t

n l e w x

Note that Mo includes only letters and writes them all in lower case. In this example, Mo used the character ‘x’ to pad the message out to make a rectangle, although he could have used any letter. Mo then sends the message to Larry by writing the letters in each row, alternating left-to-right and right-to-left. So, the above would be encrypted as

toioynnkpheleaigshareconhtomesnlewx

Your job is to recover for Larry the original message (along with any extra padding letters) from the encrypted one.

### Input

There will be multiple input sets. Input for each set will consist of two lines. The first line will contain an integer in the range 2...20 indicating the number of columns used. The next line is a string of up to 200 lower case letters. The last input set is followed by a line containing a single 0, indicating end of input.

### Output

Each input set should generate one line of output, giving the original plaintext message, with no spaces.

### Example

**Input:**

5

toioynnkpheleaigshareconhtomesnlewx

3

ttyohhieneesiaabss

0

**Output:**

theresnoplacelikehomeonasnowynightx

thisistheeasyoneab

Fiona has always loved poetry, and recently she discovered a fascinating poetical form. Tautograms are a special case of alliteration, which is the occurrence of the same letter at the beginning of adjacent words. In particular, a sentence is a tautogram if all of its words start with the same letter.

For instance, the following sentences are tautograms:

• Flowers Flourish from France

• Sam Simmonds speaks softly

• Peter pIckEd pePPers

• truly tautograms triumph

Fiona wants to dazzle her boyfriend with a romantic letter full of this kind of sentences. Please help Fiona to check if each sentence she wrote down is a tautogram or not.

# Input

Each test case is given in a single line that contains a sentence. A sentence consists of a sequence  of at most 50 words separated by single spaces. A word is a sequence of at most 20 contiguous  uppercase and lowercase letters from the English alphabet. A word contains at least one letter and a sentence contains at least one word.

The last test case is followed by a line containing only a single character ‘\*’ (asterisk).

# Output

For each test case output a single line containing an uppercase ‘Y’ if the sentence is a tautogram, or an uppercase ‘N’ otherwise.

# Sample input

Flowers Flourish from France   
Sam Simmonds speaks softly

 Peter pIckEd pePPers

truly tautograms triumph

this is NOT a tautogram

\*

### Output for the sample input

Y

Y

Y

Y

N

According to Wikipedia, an arithmetic progression (AP) is a sequence of numbers such that the difference of any two successive members of the sequence is a constant. For instance, the sequence 3, 5, 7, 9, 11, 13, . . . is an arithmetic progression with common difference 2. For this problem, we will limit ourselves to arithmetic progression whose common difference is a non-zero integer.  
On the other hand, a geometric progression (GP) is a sequence of numbers where each term after the first is found by multiplying the previous one by a fixed non-zero number called the common ratio. For example, the sequence 2, 6, 18, 54, . . . is a geometric progression with common ratio 3. For this problem, we will limit ourselves to geometric progression whose common ratio is a non-zero integer.  
Given three successive members of a sequence, you need to determine the type of the progression and the next successive member.

### Input

Your program will be tested on one or more test cases. Each case is specified on a single line with three integers (−10, 000 < a1 , a2 , a3 < 10, 000) where a1 , a2 , and a3 are distinct.  
The last case is followed by a line with three zeros.

### Output

For each test case, you program must print a single line of the form:  
XX v  
where XX is either AP or GP depending if the given progression is an Arithmetic or Geometric Progression. v is the next member of the given sequence. All input cases are guaranteed to be either an arithmetic or geometric progressions.

### Example

**Input:**  
4 7 10  
2 6 18  
0 0 0  
  
**Output:**  
AP 13  
GP 54

Contest Ranks  
  
Do you know how teams are sorted in the ACM programming contest . It is very fun , The teams are first sorted according to the number of problems they solve , when there is a tie  
the teams are sorted according to their total time , the lower your time the higher your rank .   
The time is calculated by adding to your total time the time you have taken to solve the problem when you got accepted , and their is a 20 minutes penality added to your time for  
every wrong answer you get if and only if your problem is finally accepted .  
ACM organization make a very big contest todayand they need a program to sort the teams and print them in a sorted order ?  
  
Input :   
the first line contains an interger T (number of test cases) , Each test case begins with an integer N (nunmber of teams) , following there are N lines each contains 6 integers separated  
by a single space T1 T2 T3 R1 R2 R3 , T1 means number of minutes taken to solve the first problem , T2 means number of minutes taken to solve the second problem ,   
T3 means number of minutes taken to solve the third problem , R1 means number of trials taken to solve first problem ,R2 means number of trials taken to solve second problem ,  
R3 means number of trials taken to third problem .  
  
Output  
for each test case print N lines , represents the team number in sorted order , Print an empty line between test case.  
  
Test cases  
2  
3  
0 100 200 4 1 4  
200 100 600 1 5 2  
2 400 0 0 1 2 0  
4  
0 777 0 3 0 7  
116 117 36 3 1 1  
56 200 300 1 4 5  
44 96 0 3 1 2  
  
Output :  
2   
0   
1   
  
0   
3   
2   
1