

Problem A. Collision Course

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 64 megabytes

Two cars are moving on a straight road of length l meters. At the current moment, the first car is at position p_1 meter (from the start of the road) with a velocity of v_1 meters per second. Similarly, the second car is at position p_2 meter with a velocity of v_2 meters per second. For some magical reason, both of these car cannot change their velocity. With these information, determine if the two car will eventually collide with each other.

Input

The first line consist of four integers p_1, v_1, p_2 and v_2 ($-10^9 \leq p_1, v_1, p_2, v_2 \leq 10^9$). If the velocity is negative, that means the car is moving backwards. The position of the two car will never be the same.

Output

Print “Collision” if the two car will collide in the future, or “Safe” if the two car will never collide.

Example

standard input	standard output
10 1 20 1	Safe

Problem B. Hexa Addition

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 megabytes

Once upon a time, there was a student named Talal. Talal is very hard working young man and he likes to add numbers in different numeral system. He knows how to add two decimal numbers, for example $22 + 5 = 27$. Furthermore, he knows how to add two binary numbers, for instance $10 + 11 = 101$. However, he stumbles upon the addition of two numbers in hexadecimal.

Talal is a friend of yours, and he came to you and ask for help! Can you help him?!

You are given two hexadecimal number, you are required to output their addition in hexadecimal.

Input

The input consist of two lines. The first line contains the first hexadecimal value. The second line contains the second hexadecimal value. The value could be as large as 15 digits hexadecimal value and as low as 1 digit hexadecimal value. You can assume that the Input hexadecimal values are all in uppercase.

Output

The output consist of a single line which should display the summation of the two hexadecimal values in hexadecimal.

Examples

standard input	standard output
3 3	6
F F	1E

Problem C. All About Those Primes

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **64 megabytes**

A prime number is a natural positive number greater than 1 which is only divisible by 1 and itself. 7 is a prime number because it is only divisible by 1 and 7. The only even prime number is 2, because the rest of the even numbers are divisible by 2.

Prime numbers are infinite, so far the largest prime number that we know of today (Jul, 2016) is $2^{74,207,281} - 1$ and it contains 22,338,618 digits. Its a very big family of prime numbers. Some of them are also considered twins in a weird way.

Two prime numbers are twin if their gap is 2, meaning 5 and 7 are twins as $|5-7| = 2$. Isn't that interesting, you are welcome to read about it further on your own. There is an open ended question in the Mathematics community on weather the number of twin primes are also infinite or not. But we are not going that deep into this big question yet.

Your task is simple, given a number n , you have to find out for all numbers from 1 to n if they have a twin prime or not.

Input

The only input is n ($1 \leq n \leq 1000$).

Output

If the number is a prime number with a twin then print "Twin", if no twin then print "Lonely Prime" or else if the number is not a prime print "Ordinary" without the quotes one at a line.

Example

standard input	standard output
5	Ordinary Lonely Prime Twin Ordinary Twin

Problem D. I Need My Nap!

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 64 megabytes

According to National Sleep Foundation more than 85% of mammalian species are polyphasic sleepers, meaning that they sleep for short periods throughout the day. Humans are part of the minority of monophasic sleepers, meaning that our days are divided into two distinct periods, one for sleep and one for wakefulness. It is not clear that this is the natural sleep pattern of humans. Young children and elderly persons nap, for example, and napping is a very important aspect of many cultures. But unfortunately nowadays because of our busy schedules it's quite difficult to find time for a nap. Therefore you as a programmer are assigned to design a program to read one's daily schedules and meetings throughout the day and let him know the longest nap he can have during the day and what time the nap should start.

Input

The input will consist on an arbitrary number of test cases, each test case represents one day. The first line of each set contains a positive integer (not greater than 100) representing the number of scheduled appointments during that day. In the next lines there are the appointments in the following format:

time1 time2 appointment

Where time1 represents the time, which the appointment starts and time2 the time it ends. All times will be in the 'hh:mm' format, time1 will always be strictly less than time2, they will be separated by a single space and all times will be greater than or equal to 10:00 and less than or equal to 18:00.

So, your response must be in this interval as well (i.e. no nap can start before 10:00 and last after 18:00). The appointment can be any sequence of characters, but will always be in the same line. You can assume that no line will be longer than 255 characters, that $10 \leq hh \leq 18$ and that $0 \leq mm < 60$. The scheduled appointments are in the increasing order. You must read the input until you reach the end of file.

Output

For each test case, you must print the following line:

Day d : the longest nap starts at hh : mm and will last for [H hours and] M minutes.

Where d stands for the number of the test case (starting from 1) and hh:mm is the time when the nap can start. To display the duration of the nap, follow these simple rules:

1. If the total duration X in minutes is less than 60, just print 'M minutes', where $M = X$.
2. If the total duration X in minutes is greater or equal to 60, print 'H hours and M minutes', where $H = X/60$ (integer division, of course) and $M = X$

The duration of the nap is calculated by the difference between the ending time free and the beginning time free. That is, if an appointment ends at 14:00 and the next one starts at 14:47, then you have $(14:47) - (14:00) = 47$ minutes of possible nap. If there is more than one longest nap with the same duration, print the earliest one. You can assume that there will be at least one possible nap.

Example

standard input	standard output
4 10:00 12:00 Lectures 12:00 13:00 Lunch, like always. 13:00 15:00 Boring lectures... 15:30 17:45 Reading	Day 1: the longest nap starts at 15:00 and will last for 30 minutes.

Problem E. Zrog the Frog

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 megabytes

Everybody doesn't know about Zrog the frog, because most frog don't understand what he writes down. As he doesn't like to talk much, he writes a lot. Unfortunately the other frogs have a hard time identifying which statements are from Zrog and which are not. Fortunately we got to know from his close friend Trog that he writes in mathematical expressions (Mostly because he likes Zebras with A1 in the prefix). According to Trog, Zrog follows the following rules while writing his thoughts using mathematical expressions –

- It always only consists of two sides, Left Hand Side (LHS) and Right Hand Side (RHS).
- LHS and RHS will be divided only with these symbols: '=', '<', '>', '<=' and '>='
- LSH and RHS must contain at least one or more operands separated by arithmetic operators.
- Following symbols are considered as operator and operands –
 - Operators: '+', '-', '*', '/' and '%'
 - Operands: Letters 'a' to 'z' (case insensitive) and Digits 0 to 9
- Each part of the expression is separated by a space.

Help the frog community to figure out which statements are actually from Zrog.

Input

The first line of input contains one integer n ($0 \leq n \leq 100$) which is the total number of statements. Then the following n lines contain one statement each of maximum length 100.

Output

For each statement print "Zrog" if it is an expression according to the above rules or else print "Frog" without the quotes.

Examples

standard input	standard output
5 3 + 4 = 7 ab + cd < e + f This statement is from Zrog. a * b + c = 12 X + Y =< Z	Zrog Zrog Frog Zrog Frog
3 Hat / Lac = 6 / f 8 / 53 / 9 * 798 + k - X < 6 * gka	Zrog Frog Zrog