**Problem A**

**10071 - Back to High School Physics**

Time limit: 3.000 seconds

A particle has initial velocity and acceleration. If its velocity after certain time is v then what will its displacement be in twice of that time?

**Input**

The input will contain two integers in each line. Each line makes one set of input. These two integers denote the value of *v* (*-*100 *<= v <=*100) and *t* (0 *<= t <=*200) (*t* means at the time the particle gains that velocity)

**Output**

For each line of input print a single integer in one line denoting the displacement in double of that time.

**Sample Input**

0 0

5 12

**Sample Output**

0

120

**Problem B**

**113 - Power of Cryptography**

Time limit: 3.000 seconds

Current work in cryptography involves (among other things) large prime numbers and computing powers of numbers modulo functions of these primes. Work in this area has resulted in the practical use of results from number theory and other branches of mathematics once considered to be of only theoretical interest.

This problem involves the efficient computation of integer roots of numbers.

## The Problem

Given an integer tex2html_wrap_inline32 and an integer tex2html_wrap_inline34 you are to write a program that determines tex2html_wrap_inline36 , the positive tex2html_wrap_inline38 root of *p*. In this problem, given such integers *n* and *p*, *p* will always be of the form tex2html_wrap_inline48 for an integer *k* (this integer is what your program must find).

## The Input

The input consists of a sequence of integer pairs *n* and *p* with each integer on a line by itself. For all such pairs tex2html_wrap_inline56 ,tex2html_wrap_inline58 and there exists an integer *k*, tex2html_wrap_inline62 such that tex2html_wrap_inline64 .

## The Output

For each integer pair *n* and *p* the value tex2html_wrap_inline36 should be printed, i.e., the number *k* such that  tex2html_wrap_inline64 .

## Sample Input

2

16

3

27

7

4357186184021382204544

## Sample Output

4

3

1234

**Problem C**

### 11172 - Relational Operator

Time limit: 3.000 seconds

Some operators checks about the relationship between two values and these operators are called relational operators. Given two numerical values your job is just to find out the relationship between them that is (i) First one is greater than the second (ii) First one is less than the second or (iii) First and second one is equal.

##### **Input**

First line of the input file is an integer t (t<15) which denotes how many sets of inputs are there. Each of the next t lines contain two integers a and b (|a|,|b|<1000000001).

#### Output

For each line of input produce one line of output. This line contains any one of the relational operators “>”, “<” or “=”, which indicates the relation that is appropriate for the given two numbers.

# Sample Input                           Output for Sample Input

|  |  |
| --- | --- |
| 3  10 20  20 10  10 10 | <  >  = |

**Problem D**

### 579 - Clock Hands

Time limit: 3.000 seconds

The history of clocks is fascinating, but unrelated to this problem. In this problem, you are asked to find the angle between the minute hand and the hour hand on a regular analog clock. Assume that the second hand, if there were one, would be pointing straight up at the 12. Give all angles as the smallest positive angles. For example 9:00 is 90 degrees; not -90 or 270 degrees.

## Input

The input is a list of times in the form *H*:*M*, each on their own line, with $1 \le H \le 12$ and $00 \le M \le 59$. The input is terminated with the time 0:00. Note that *H* may be represented with 1 or 2 digits (for 1-9 or 10-12, respectively); *M* is always represented with 2 digits (The input times are what you typically see on a digital clock).

## Output

The output displays the smallest positive angle in degrees between the hands for each time. The answer should between 0 degrees and180 degrees for all input times. Display each angle on a line by itself in the same order as the input. The output should be rounded to the nearest 1/1000, i.e., three places after the decimal point should be printed.

## Sample Input

12:00

9:00

8:10

0:00

## Sample Output

0.000

90.000

175.000

**Problem E**

### 10035 - Primary Arithmetic

Time limit: 3.000 seconds

Children are taught to add multi-digit numbers from right-to-left one digit at a time. Many find the “carry” operation - in which a 1 is carried from one digit position to be added to the next - to be a significant challenge. Your job is to count the number of carry operations for each of a set of addition problems so that educators may assess their difficulty.

**Input**

Each line of input contains two unsigned integers less than 10 digits. The last line of input contains ‘0 0’.

**Output**

For each line of input except the last you should compute and print the number of carry operations that would result from adding the two numbers, in the format shown below.

**Sample Input**

123 456

555 555

123 594

0 0

**Sample Output**

No carry operation.

3 carry operations.

1 carry operation.

**Problem F**

**10055 - Hashmat the Brave Warrior**

Time limit: 3.000 seconds

Hashmat is a brave warrior who with his group of young soldiers moves from one place to another to fight against his opponents. Before Fighting he just calculates one thing, the difference between his soldier number and the opponent’s soldier number. From this difference he decides whether to fight or not. Hashmat’s soldier number is never greater than his opponent.

**Input**

The input contains two numbers in every line. These two numbers in each line denotes the number soldiers in Hashmat’s army and his opponent’s army or vice versa. The input numbers are not greater than 232. Input is terminated by ‘End of File’.

**Output**

For each line of input, print the difference of number of soldiers between Hashmat’s army and his opponent’s army. Each output should be in separate line.

**Sample Input**

10 12

10 14

100 200

**Sample Output**

2

4

100