UNIVERSIDAD DISTRITAL
FRANCISCO JOSE DE CALDAS

# III MARATON INTERNA DE PROGRAMACION

# MARATON SEMANA TECNOLOGICA

# CATEGORIA AVANZADOS

# UNIVERSIDAD DISTRITAL

# 31 de Octubre de 2007

# PROBLEMAS

**Elaborado por: Hector Florez**
**Basado de www.acis.org.co**

# Problem 1. Contest System Quality Assurance Tester

Source file name: quality.c, quality.cpp or quality.java
Input: quality.in
Output: standard output

Write a program to score a small, three-problem programming contest. Each input line contains six space-separated integers representing raw score data. The first three integers are in the range 0 . . . 100000. They represent seconds taken to solve the first, second, and third problems, respectively. Zero seconds indicates that a problem has not been solved. The last three integers are in the range 0 . . . 100, representing the attempts taken to solve the first, second, and third problems, respectively. Every failed attempt is penalized with 20 minutes, but only for problems that are eventually solved.

Each output line should begin with the string team, followed by a single space, the input line number, a colon, a single space, the number of solved problems, a comma, a single space, and the total number of seconds including penalties it took for the solved problems.

| Sample input | Sample Output |
|---|---|
| 0 777 0 4 1 1<br>1 1 1 1 1 1 | team 1: 1, 777<br>team 2: 3, 3 |

# Problem 2. Uncle Jack

Source file name: uj.c, uj.cpp or uj.java
Input: uj.in
Output: standar output

Dear Uncle Jack is willing to give away some of his collectable CDs to his nephews. Among the titles you can find very rare albums of Hard Rock, Classical Music, Reggae and much more; each title is considered to be unique. Last week he was listening to one of his favorite songs, Nobody's fool, and realized that it would be prudent to be aware of the many ways he can give away the CDs among some of his nephews.
So far he has not made up his mind about the total amount of CDs and the number of nephews. Indeed, a given nephew may receive no CDs at all.
Please help dear Uncle Jack, given the total number of CDs and the number of nephews, to calculate the number of different ways to distribute the CDs among the nephews.

## Input

The input consists of several test cases. Each test case is given in a single line of the input by, space separated, integers N (1 <= N <= 10) and D (0 <= D <= 25), corresponding to the number of nephews and the number of CDs respectively. The end of the test cases is indicated with N = D = 0.

## Output

The output consists of several lines, one per test case, following the order given by the input. Each line has the number of all possible ways to distribute D CDs among N nephews. The output must be written to standard output.

| Sample input | Sample Output |
|---|---|
| 1 20<br>3 10<br>0 0 | 1<br>59049 |

# Problem 3. Christmas Trees

Source file name: tree.c, tree.cpp or tree.java
Input: tree.in
Output: standar output

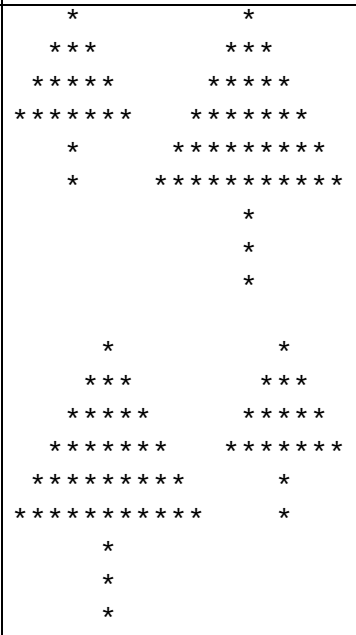
Christmas is always fun at the rebel home base. This year they are buying two Christmas trees and are placing them in elevated stands so that the tops of the trees are level if the trees are of different heights. Given the heights of the trees, you have to print a picture of the two Christmas trees.

## Input

The input consists of several lines. Each line will contain two positive decimal integers no larger than 100. These integers will be separated by exactly one space. These integers represent the sizes of the trees; the size of the left tree is followed by the size of the right tree. The last line will contain two zeros, separated by one. This line is not to be processed; it merely signifies the end of the input.

## Output

The output cases are to appear in the same order in wich they appear in the input. A full specification is not necessary; the examples will suffice, but here area few pointers: The height of the stem is the same as the input integer; the height of the branches is twice that. The tops of trees are level; the bottoms need not be. There is exactly one space between trees, wich is to say, between the trees is there is exactly one vertical column consisting only of spaces. All trailing spaces are trimmed from each line before printing; the last character of any line in the output file should not be a space. There should be exactly one blanck line following each output case.

| Sample input | Sample Output |
|---|---|
| 2 3<br>3 2<br>0 0 |  |

# Problem 4. Making Money

Source file name: money.c, money.cpp or money.java
Input: money.in
Output: standar output

A trick sometimes used by parents to teach their children the value of money is to give then a penny - just a penny! - and the promise that for each day they don't spend it, the parent will double it. All students of computing know that long before a month has elapsed without spending a cent, the parents will not likely be able to make good on their promise. 100-percent compound daily interest on an investment is, of course, unattainable in normal financial dealings, but we are all continually reminded of the power of compound interest, even with the relatively low interest rates available today.

But exactly how much money can be made with compound interest? Assume, for example, an initial investment of $100.00 (US or Canadian), an annual interest rate of 6.00 percent, and that interest is compounded monthly. That is, the interest earned during the preceding month is added to the principal at the end of the month. (For our purposes, we'll assume a month is exactly 1/12th of a year.)

At the end of the first month, the money will have earned 0.5 percent interest (1/12th of 6.00 percent), or $0.50. This is added to the $100.00 invested, so that during the next month, interest is paid on $100.50. During the next month another 0.5 percent interest is earned, which is exactly $0.5025. We will assume that the bank, being conservative, will not pay any interest less than $0.01, so our investment is credited with an additional $0.50 at the end of the second month, for a whopping total of $101.00. Continuing in the same manner, at the end of 12 months our investment will total $106.12, $0.12 more than simple 6.00 percent interest for a year with no compounding. Given an amount P to be invested for a year with I percent interest, compounded C times during the year at equal intervals, what is total return on the investment?

## Input

There will be multiple cases to consider. The input for each case is a single line containing the initial investment amount, P, given in dollars and cents (but no fractional cents, and no larger than $100,000.00), the annual interest rate (I) given as a real number with two fractional digits representing a percentage, greater than zero but less than 100, and the number of compounding intervals per year (C), an integer between 1 and 365. The last case will be followed by a line containing "0.00 0.00 0".

## Output

For each input case, display the case number (1, 2, . . . ), the initial investment (P), the annual interest rate (I), the number of compounding intervals per year(C), and the value of the investment at the end of a year. Your output should follow the format shown in the examples below.

| Sample input | Sample Output |
|---|---|
| 100.00 6.00 1<br>100.00 6.00 12<br>1000.00 6.00 12<br>0.00 0.00 0 | Case 1. $100.00 at 6.00% APR compounded 1 times yields $106.00<br>Case 2. $100.00 at 6.00% APR compounded 12 times yields $106.12<br>Case 3. $1000.00 at 6.00% APR compounded 12 times yields $1061.63 |

# Problem 5. RSA

Source file name: rsa.c, rsa.cpp or rsa.java
Input: rsa.in
Output: standar output

One of the methods of cryptography used at present, is algorithm RSA, this system is based on the difficulty of the mathematical problem of the factoring of a compound number, for what algorithms of time do not exist polinomial, and in the facility of the inverse operation, to multiply it.

The basic functioning becomes considering the variables p, q, e and M where M is the Message that must enter and change to a Hexadecimal output.
The conditions to validate the information are:

- p and q must be prime numbers.
- The message must not have a limit of information and / or characters.
- The exit must be only hexadecimal.
- n = p * q
- Φ: Φ (n)=(p-1)(q-1)
- Find a number e that the MCD(e, Φ (n))=1.
- You must calculate d=((Y* Φ(n))+1)/e for Y=1,2,3,... until d becomes integer.
- Numbers (e, d) are the public key
- Numbers (d, n) are the private key
- The encrypted message is obtained by the formula: C = (M^e) mod n
- The original message is obtained by the formula: M = (C^d) mod n

## Input

The entry consists of a several cases. Each one of cases has the number p, number q, number e and the message M.
The coding will be realized applying the formula: C = (M^e) mod n

## Output

The output is determined by the encrypted message coded in Hexadecimal.

| Sample input | Sample Output |
|---|---|
| 53 23 15 Hello<br>37 19 5 RSA TEST | 01EF00A703FD03FD00CC<br>02B100B6017500F2022000E900D200A5 |

# Problem 6. Hello Kitty

Source file name: hkitty.c, hkitty.cpp or hkitty.java
Input: hkitty.in
Output: standar output

Kitty sends a kind of original email messages to her friend Garf. To write a message, she chooses a word W and a number n and replicates W n times horizontally. Then she repeats this string in the next line, but rotating the characters once to the left. And she repeats this 'rotateand-output' process until the word W appears displayed as the first column of the rectangular pattern that she produces.

As an example, when she chooses the word Hello and the number 3, she gets the pattern:

```
HelloHelloHello
elloHelloHelloH
lloHelloHelloHe
loHelloHelloHel
oHelloHelloHell
```

Kitty has been sending such emails during the last three years. Recently, Garf told her that perhaps her work may be automatized with a software to produce Kitty's patterns. Could you help her?

## Input

The input file contains several test cases, each one of them in a separate line. Each test case has a word and a positive integer that should generate the corresponding rectangular pattern. The word is a string of alphabetic characters (a..z). The number is less than 10. A line whose contents is a single period character means the end of the input (this last line is not to be processed).

## Output

Output texts for each input case are presented in the same order that input is read. For each test case the answer must be a left aligned Kitty pattern corresponding to the input.

| Sample input | Sample Output |
|---|---|
| Love 1<br>Kitty 2<br>. | Love<br>oveL<br>veLo<br>eLov<br>KittyKitty<br>ittyKittyK<br>ttyKittyKi<br>tyKittyKit<br>yKittyKity |