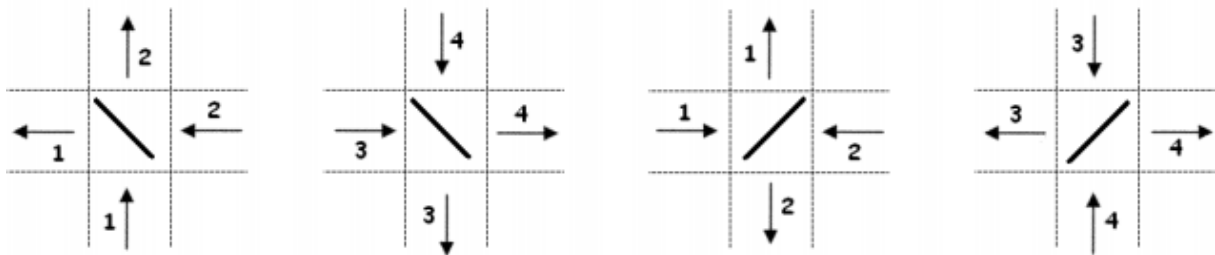


Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 24 февруари 2013 г.

А. Лабиринт

Провеждаме експеримент, пускайки светлинен лъч в лабиринт (за простота представен чрез матрица $N \times M$ ($1 \leq N, M \leq 500$)). Всяка клетка може да бъде празна (“.”), черна дупка (“С”) или огледало, означено с един от символите “/” или “\”, в зависимост от това накъде е наклонено. Когато светлината попада върху огледало се отклонява по един от следните начини:



Светлината може да се разпространява по редове и колони, като когато достигне до огледало, променя посоката си на 90 градуса, спазвайки горепоказаните правила. Преминаването през една клетка до нейна съседна става за 1 секунда. Ако светлината достигне до черна дупка, бива погълната. Светлината „напуска лабиринта“, ако достигне до някоя гранична клетка.

Вашата задача е по зададена стартова позиция, от която пускаме светлината, да определите максималния период от време (в секунди), през който светлината ще се движи в лабиринта, преди да го напусне или да бъде погълната от черна дупка.

На първия ред на всеки тест са зададени N и M . Следват N реда с по M символа, описващи лабиринта. Последният ред от примера съдържа стартовата позиция (ред, колона). Индексирането започва от 1. Краят на входа е маркиран с “END”.

За всеки тест на първия ред извеждайте избраната начална посока - “U” за нагоре, “R” за надясно, “D” за надолу и “L” за наляво, в която трябва да насочим лъча. Втория ред трябва да съдържа търсеното време или „Forever“ ако движението на светлината може да продължи безкрайно дълго време. Ако съществуват повече от едно възможни най-добри времена и различни посоки, то избирайте посоката, базирайки се на следния приоритет: “U”, “R”, “D”, “L”. Извеждайте по един празен ред между резултатите от всеки два тестови примера.

Вход	Продължение на входа	Изход
5 5 ../.\C... ...C. \.../ 3 3	5 7 /.....\ ../....\ \...../ /.....\ \....../ 3 3 END	U 17 R Forever



Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 24 февруари 2013 г.

В. Еднаквост

Дадена е редица от N ($2 \leq N \leq 20$) цели положителни числа. От текущата редица на всяка итерация получаваме нова, като $a'_k = |a_k - a_{k+1}|$, започвайки последователно от първия ѝ елемент и $a'_N = |a_N - a_1|$. Колко итерации са нужни, за да получим редица от еднакви елементи?

За $N = 4$ и редицата 0 2 5 11 са нужни следните 8 итерации:

```

2 3 6 11
1 3 5 9
2 2 4 8
0 2 4 6
2 2 2 6
0 0 4 4
0 4 0 4
4 4 4 4

```

Вход

Всеки тестов пример се състои от един ред, на който с по един празен символ са разделени елементите на редицата.

Изход

За всеки тестов пример, на отделен ред на стандартния изход, изведете търсения брой итерации във формата, показан в примерите по-долу. Ако след 1000 итерации все още не се получава желаната редица от еднакви числа изведете „not attained“ за съответния тест.

Вход	Изход
0 2 5 11	Case 1: 8 iterations
0 2 5 11 3	Case 2: not attained
0 4 0 4	Case 3: 1 iteration
300 8600 9000 4000	Case 4: 3 iterations
1 1 1	Case 5: 0 iterations



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 24 февруари 2013 г.

C. Speed Racer

Speed Racer must go go go rescue Trixie at the top of Mount Domo! He must get there as quickly as possible, but his Mach 5 only holds a specific amount of fuel, and there is no way to refuel on the way. Luckily, he knows precisely how much fuel is consumed at a particular speed, taking into account air resistance, tire friction, and engine performance. For a given speed v in kilometers per hour, the amount of fuel consumed in liters per hour is

$$av^4 + bv^3 + cv^2 + dv$$

Assuming Speed travels at a constant speed, his tank holds t liters of fuel, and the top of Mount Domo is m kilometers away, how fast must he drive?

Input

The input will be one problem per line. Each line will contain six nonnegative floating point values representing a , b , c , d , m , and t , respectively. No input value will exceed 1000. The values of c , d , m , and t will be positive. There will always be a solution.

The output should be formatted as a decimal number with exactly two digits after the decimal point, and no leading zeros. The output value should be such that the Speed Racer will not run out of fuel (so truncate, rather than round, the final result). You are guaranteed that no final result will be within 10^{-6} of an integer multiple of 0.01.

Output

The required output will be a single floating point value representing the maximum speed in kilometers per hour that Speed Racer can travel to reach the top of Mount Domo without running out of gas.

Sample Input	Sample Output
0.000001 0.0001 0.029 0.2 12 100	134.41
2.8e-8 7.6e-6 0.0013 0.47 11.65 20.81	257.45
1.559e-7 1.8195e-5 0.0022233 0.31292 58.902 85.585	142.65



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 24 февруари 2013 г.

Е. Вендинг машина

Продукти, предлагани от автомат се продават за цяло число левове. Автоматът връща ресто под формата на монети от един лев, както и банкноти от по пет и десет лева. Напишете програма за автомата като са спазват следните две условия: При всяка отделна покупка рестото да се съдържа по-малко от 5 монети, и по-малко от две банкноти от 5 лева.

Вход

Всеки отделен ред на стандартния вход описва един тестов пример, съдържащ две цели положителни числа. Първото е стойността на покупката, а второто – парите въведени в автомата. Края на входа е ред, съдържащ две нули.

Изход

За всеки тестов пример извеждайте по един отделен ред, състоящ се от: номера на поредния тест, рестото, броят на десет и пет левовите банкноти, както и този на монетите от един лев, върнати от автомата за конкретния пример. Изходът да бъде форматиран, както е показано по-долу.

Вход	Изход
72 100	Case 1: 28 = 2 * 10 + 1 * 5 + 3 * 1
37 200	Case 2: 163 = 16 * 10 + 0 * 5 + 3 * 1
5 50	Case 3: 45 = 4 * 10 + 1 * 5 + 0 * 1
0 0	



Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 24 февруари 2013 г.

Е. Средно аритметично

На всеки отделен не празен ред на стандартния вход ще бъдат зададени не повече от 500 измежду символите "A", "B", "C", "D" или "F", разделени с по не повече от 3 интервала. Всеки от тях носи съответно по 4, 3, 2, 1 и 0 точки. За всеки тестов пример, изведете на отделен ред, средното аритметично на точките на символите с точност 2 знака, в показания по-долу формат:

Вход	Изход
A B C D F	Case 1: 2.00
B F F C C A	Case 2: 1.83
D C E F	Case 3: Unknown letter grade in input

Забележка: Ако някой от непразните символи е различен от изброените по-горе символи, то изведете "Unknown letter grade in input".



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 24 февруари 2013 г.

G. Nim-B* Sum

This problem has nothing to do with siting sewage plants, power lines or wind farms. NIM is an ambigram .

The game of NIM is played with any number of piles of objects with any number of objects in each pile. At each turn, a player takes one or more (up to all) objects from one pile. In the normal form of the game, the player who takes the last object is the winner. There is a well-known strategy for this game based on the nim-2 sum.

The Nim-B sum (nim sum base **B**) of two non-negative integers **X** and **Y** (written *NimSum(B, X, Y)*) is computed as follows:

1. Write each of **X** and **Y** in base **B**.
2. Each digit in base **B** of the Nim-B sum is the sum modulo **B** of the corresponding digits in the base **B** representation of **X** and **Y**.

For example:

$$\begin{aligned} \text{NimSum}(2, 123, 456) &= 1111011 \text{ } \times \text{ } 111001000 = 110110011 = 435 \\ \text{NimSum}(3, 123, 456) &= 11120 \text{ } \times \text{ } 121220 = 102010 = 300 \\ \text{NimSum}(4, 123, 456) &= 1323 \text{ } \times \text{ } 13020 = 10303 = 307 \end{aligned}$$

The strategy for normal form Nim is to compute the Nim-2 sum **T** of the sizes of all piles. If at any time, you end your turn with **T** = 0, you are guaranteed a **WIN**. Any opponent move must leave **T** not 0 and there is always a move to get **T** back to 0. This is done by computing *NimSum(2, T, PS)* for each pile; if this is less than the pile size (PS), compute the difference between the PS and the Nim-2 sum and remove it from that pile as your next move.

Write a program to compute NimSum(B, X, Y).

The first line of **input** contains a single integer **P**, ($1 \leq P \leq 1000$), which is the number of data sets that follow. Each data set is a single line that contains the data set number, followed by a space, followed by three space separated decimal integers, **B**, **X** and **Y**. $2 \leq B \leq 2000000$, $0 \leq X \leq 2000000$, $0 \leq Y \leq 2000000$.

For each data set there is one line of **output**. It contains the data set number followed by a single space, followed by **N**, the decimal representation of the Nim sum in base **B** of **X** and **Y**.

Sample Input	Sample Output
4	1 435
1 2 123 456	2 300
2 3 123 456	3 307
3 4 123 456	4 429
4 5 123 456	



Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 24 февруари 2013 г.

H. Alien Communicating Machines

Aliens on the planet Hex have sixteen fingers. As a result, they have developed a system of writing numbers with sixteen digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, in ascending order). Like the decimal system that we use, a number is written as a sequence of digits, and the value of a number is determined by the following rules:

- The value of a single-digit number is just the value of that digit.
- The value of number a consisting of a number b followed by a digit d is the value of b multiplied by a fixed base, plus the value of d . In our decimal system, the base is ten, whereas in the system used by the Hex aliens, the base is sixteen.

Still other planets have other aliens with different numbers of fingers. These aliens use the same rules for writing numbers, but each uses a different base.

You have the important job of promoting universal peace by improving communication among all of these aliens. Specifically, you are to write a program that translates numbers written by one alien into numbers that can be understood by a different alien.

Input Specification

The first line of input contains a single integer, the number of test cases to follow. Each test case is a single line containing three numbers x , y , and z . Both x and y are written in decimal, and are the bases used by the two aliens. Each is at least two and at most thirty-six. The number z is written in base x , using the letters A through Z as the digits with value 10 through 35. The number z will be no greater than four billion.

Sample Input

```
1
16 10 11
```

Output Specification

For each line of input, output a line containing a single number, equal in value to z , but written in base y .

Output for Sample Input

```
17
```




Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 24 февруари 2013 г.

I. Pills

Aunt Lizzie takes half a pill of a certain medicine every day. She starts with a bottle that contains N pills.

On the first day, she removes a random pill, breaks it in two halves, takes one half and puts the other half back into the bottle.

On subsequent days, she removes a random piece (which can be either a whole pill or half a pill) from the bottle. If it is half a pill, she takes it. If it is a whole pill, she takes one half and puts the other half back into the bottle.

In how many ways can she empty the bottle? We represent the sequence of pills removed from the bottle in the course of $2N$ days as a string, where the i -th character is W if a whole pill was chosen on the i -th day, and H if a half pill was chosen ($0 \leq i < 2N$). How many different valid strings are there that empty the bottle?

Input

The input will contain data for at most 1000 problem instances. For each problem instance there will be one line of input: a positive integer $N \leq 30$, the number of pills initially in the bottle. End of input will be indicated by 0.

Output

For each problem instance, the output will be a single number, displayed at the beginning of a new line. It will be the number of different ways the bottle can be emptied.

Sample Input	Output for Sample Input
6	132
1	1
4	14
2	2
3	5
30	3814986502092304
0	



Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 24 февруари 2013 г.

J. Non-negative Partial Sums

You are given a sequence of n numbers $a_0 \leq \dots \leq a_{n-1}$. A cyclic shift by k positions ($0 \leq k \leq n - 1$) results in the following sequence: $a_k \leq a_{k+1} \leq \dots \leq a_{n-1} \leq a_0 \leq a_1 \leq \dots \leq a_{k-1}$. How many of the n cyclic shifts satisfy the condition that the sum of the first i numbers is greater than or equal to zero for all i with $1 \leq i \leq n$?

Input

Each test case consists of two lines. The first contains the number n ($1 \leq n \leq 10^6$), the number of integers in the sequence. The second contains n integers $a_0 \leq \dots \leq a_{n-1}$ ($-1000 \leq a_i \leq 1000$) representing the sequence of numbers. The input will finish with a line containing 0.

Output

For each test case, print one line with the number of cyclic shifts of the given sequence which satisfy the condition stated above.

Sample Input

```
3
2 2 1
3
-1 1 1
1
-1
0
```

Sample Output

```
3
2
0
```