Root :: Programming Challenges (Skiena & Revilla) :: Chapter 5

https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=33

Problemas:

Código em C/C+++

Discentes	**Problemas**
BRIHAN	10105 - Polynomial Coefficients
CARLOS	10127 - Ones
DANIEL	10035 - Primary Arithmetic
JOÃO GABRIEL	10077 - The Stern-Brocot Number System
JOÃO HENRIQUE	10202 - Pairsumonious Numbers
MATEUS	847 - A Multiplication Game
PEDRO	10018 - Reverse and Add
VINICIUS	701 - The Archeologists' Dilemma

10105 Polynomial coefficients

The problem is to calculate the coefficients in expansion of polynomial $(x_1 + x_2 + \ldots + x_k)^n$.

Input

The input will consist of a set of pairs of lines. The first line of the pair consists of two integers n and k separated with space $(0 < K, N < 13)$. This integers define the power of the polynomial and the amount of the variables. The second line in each pair consists of k non-negative integers $n_1, \ldots, n_k$, where $n_1 + \ldots + n_k = n$.

Output

For each input pair of lines the output line should consist one integer, the coefficient by the monomial $x_1^{n_1} x_2^{n_2} \ldots x_k^{n_k}$ in expansion of the polynomial $(x_1 + x_2 + \ldots + x_k)^n$.

Sample Input

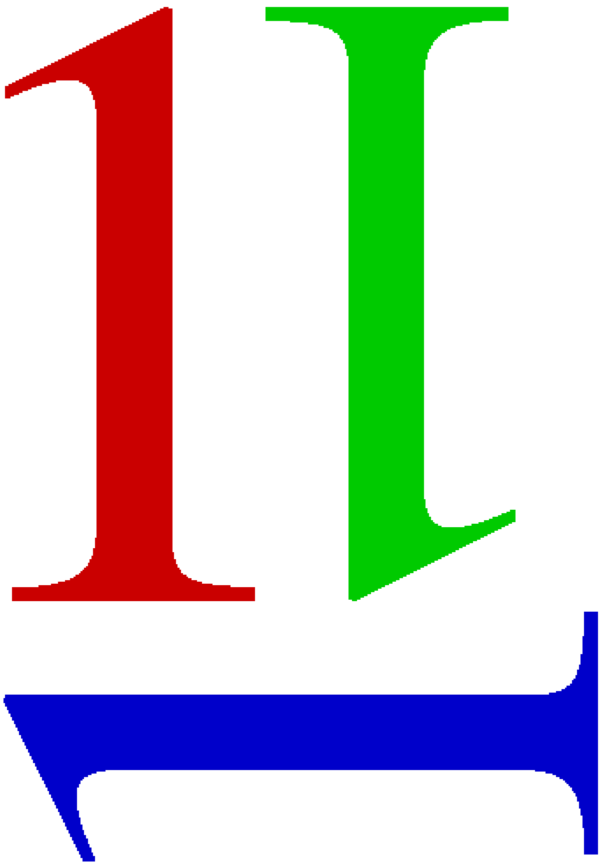
```
2 2
1 1
2 12
1 0 0 0 0 0 0 0 0 0 1 0
```

Sample Output

```
2
2
```

10127 Ones

Given any integer $0 \leq n \leq 10000$ not divisible by 2 or 5, some multiple of n is a number which in decimal notation is a sequence of 1's. How many digits are in the smallest such a multiple of n?



Input

A file of integers at one integer per line.

Output

Each output line gives the smallest integer $x > 0$ such that $p = \sum_{i=0}^{x-1} 1 \times 10^i = a \times b$, where a is the corresponding input integer, and b is an integer greater than zero.

Sample Input

```
3
7
9901
```

Sample Output

```
3
6
12
```

10035 Primary Arithmetic

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.
```

10077 The Stern-Brocot Number System

The *Stern-Brocot tree* is a beautiful way for constructing the set of all nonnegative fractions $\frac{m}{n}$ where m and n are relatively prime. The idea is to start with two fractions $\left(\frac{0}{1}, \frac{1}{0}\right)$ and then repeat the following operations as many times as desired:

Insert $\frac{m+m'}{n+n'}$ between two adjacent fractions $\frac{m}{n}$ and $\frac{m'}{n'}$.
For example, the first step gives us one new entry between $\frac{0}{1}$ and $\frac{1}{0}$,

$$\frac{0}{1}, \frac{1}{1}, \frac{1}{0};$$

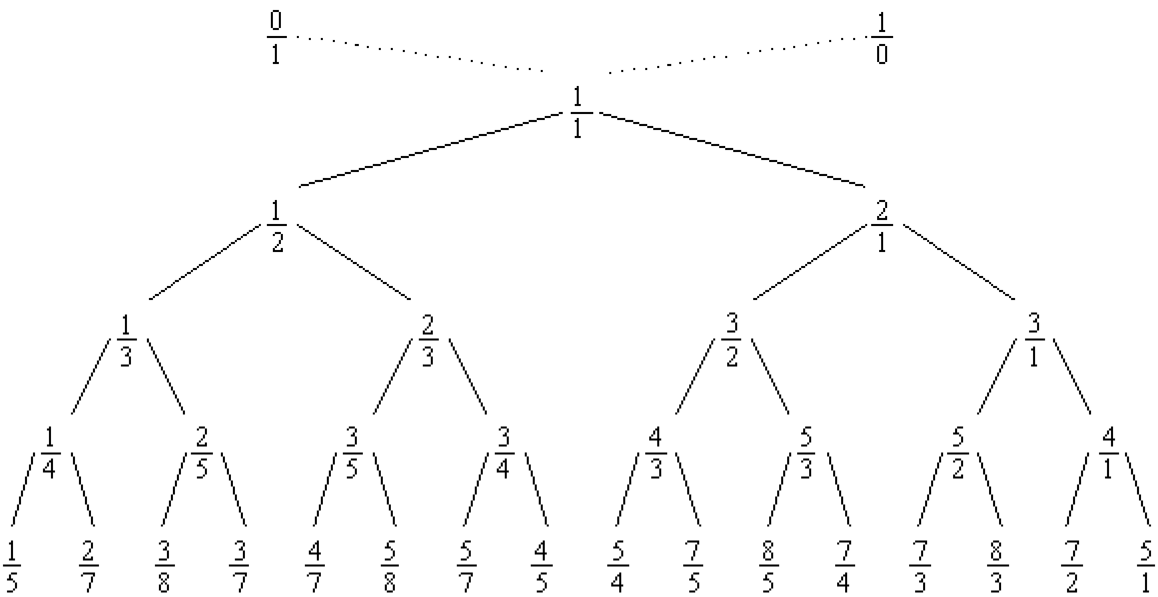
and the next gives two more:

$$\frac{0}{1}, \frac{1}{2}, \frac{1}{1}, \frac{2}{1}, \frac{1}{0}.$$

The next gives four more,

$$\frac{0}{1}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{1}{1}, \frac{3}{2}, \frac{2}{1}, \frac{3}{1}, \frac{1}{0};$$

and then we will get 8, 16, and so on. The entire array can be regarded as an infinite binary tree structure whose top levels look like this:



The construction preserves order, and we couldn't possibly get the same fraction in two different places.

We can, in fact, regard the *Stern-Brocot tree* as a *number system* for representing rational numbers, because each positive, reduced fraction occurs exactly once. Let's use the letters '`L`' and '`R`' to stand for going down to the left or right branch as we proceed from the root of the tree to a particular fraction; then a string of `L`'s and `R`'s uniquely identifies a place in the tree. For example, `LRRL` means that we go left from $\frac{1}{1}$ down to $\frac{1}{2}$, then right to $\frac{2}{3}$, then right to $\frac{3}{4}$, then left to $\frac{5}{7}$. We can consider LRRL to be a representation of $\frac{5}{7}$. Every positive fraction gets represented in this way as a unique string of `L`'s and `R`'s.

Well, actually there's a slight problem: The fraction $\frac{1}{1}$ corresponds to the empty string, and we need a notation for that. Let's agree to call it I, because that looks something like 1 and it stands for "identity".

In this problem, given a positive rational fraction, you are expected to represent it in *Stern-Brocot number system*.

Input

The input file contains multiple test cases. Each test case consists of a line contains two positive integers m and n where m and n are relatively prime. The input terminates with a test case containing two 1's for m and n, and this case must not be processed.

Output

For each test case in the input file output a line containing the representation of the given fraction in the *Stern-Brocot number system*.

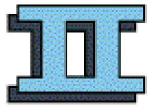
Sample Input

```
5 7
878 323
1 1
```

Sample Output

```
LRRL
RRLRRLRLLLLRLRRR
```

10202 Pairsumonious Numbers



For $10 > N > 2$ numbers we form $N * (N - 1)/2$ sums by adding every pair of the numbers. Your task is to find the N numbers given the sums.

Input

Each line of input contains N followed by $N * (N - 1)/2$ integer numbers separated by a space.

Output

For each line of input, output one line containing N integers in non-descending order such that the input numbers are pairwise sums of the N numbers. If there is more than one solution, any one will do; if there is no solution, print '`Impossible`'.

Sample Input

```
3 1269 1160 1663
3 1 1 1
5 226 223 225 224 227 229 228 226 225 227
5 216 210 204 212 220 214 222 208 216 210
5 -1 0 -1 -2 1 0 -1 1 0 -1
5 79950 79936 79942 79962 79954 79972 79960 79968 79924 79932
```

Sample Output

```
383 777 886
Impossible
111 112 113 114 115
101 103 107 109 113
-1 -1 0 0 1
39953 39971 39979 39983 39989
```

847 A multiplication game

Stan and Ollie play the game of multiplication by multiplying an integer p by one of the numbers 2 to 9. Stan always starts with $p = 1$, does his multiplication, then Ollie multiplies the number, then Stan and so on. Before a game starts, they draw an integer $1 < n < 4294967295$ and the winner is who first reaches $p \geq n$.

Input and Output

Each line of input contains one integer number n. For each line of input output one line either

```
Stan wins.
```

or

```
Ollie wins.
```

assuming that both of them play perfectly.

Sample input

```
162
17
34012226
```

Sample Output

```
Stan wins.
Ollie wins.
Stan wins.
```

10018 Reverse and Add

The "*reverse and add*" method is simple: choose a number, reverse its digits and add it to the original. If the sum is not a palindrome (which means, it is not the same number from left to right and right to left), repeat this procedure.

For example:

```
 195   Initial number
 591
 ----
 786
 687
 ----
1473
3741
 ----
5214
4125
 ----
9339   Resulting palindrome
```

In this particular case the palindrome '9339' appeared after the 4th addition. This method leads to palindromes in a few step for almost all of the integers. But there are interesting exceptions. 196 is the first number for which no palindrome has been found. It is not proven though, that there is no such a palindrome.

You must write a program that give the resulting palindrome and the number of iterations (additions) to compute the palindrome.

You might assume that all tests data on this problem:

- will have an answer ,
- will be computable with less than 1000 iterations (additions),
- will yield a palindrome that is not greater than 4,294,967,295.

Input

The first line will have a number N ($0 < N \leq 100$) with the number of test cases, the next N lines will have a number P to compute its palindrome.

Output

For each of the N tests you will have to write a line with the following data : $minimum_number_of_iterations(additions)_to_get_to_the_palindrome$ and $the_resulting_palindrome_itself$ separated by one space.

Sample Input

```
3
195
265
750
```

Sample Output

```
4 9339
5 45254
3 6666
```

701 The Archeologists' Dilemma

An archeologist seeking proof of the presence of extraterrestrials in the Earth's past, stumbles upon a partially destroyed wall containing strange chains of numbers. The left-hand part of these lines of digits is always intact, but unfortunately the right-hand one is often lost by erosion of the stone. However, she notices that all the numbers with all its digits intact are powers of 2, so that the hypothesis that all of them are powers of 2 is obvious. To reinforce her belief, she selects a list of numbers on which it is apparent that the number of legible digits is strictly smaller than the number of lost ones, and asks you to find the smallest power of 2 (if any) whose first digits coincide with those of the list.

Thus you must write a program such that given an integer, it determines (if it exists) the smallest exponent E such that the first digits of 2^E coincide with the integer (remember that more than half of the digits are missing).

Input

It is a set of lines with a positive integer N not bigger than 2147483648 in each of them.

Output

For every one of these integers a line containing the smallest positive integer E such that the first digits of 2^E are precisely the digits of N, or, if there is no one, the sentence '`no power of 2`'.

Sample Input

```
1
2
10
```

Sample Output

```
7
8
20
```