3n+1 Problem

Problems in Computer Science are often classified as belonging to a certain class of problems (e.g., NP, Unsolvable, Recursive). In this problem you will be analyzing a property of an algorithm whose classification is not known for all possible inputs.

Consider the following algorithm:

1. input n
2. print n
3. if $n = 1$ then STOP
4. if n is odd then $n \leftarrow 3n + 1$
5. else $n \leftarrow n/2$
6. GOTO 2

Given the input 22, the following sequence of numbers will be printed

22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

It is conjectured that the algorithm above will terminate (when a 1 is printed) for any integral input value. Despite the simplicity of the algorithm, it is unknown whether this conjecture is true. It has been verified, however, for all integers n such that $0 < n < 1,000,000$ (and, in fact, for many more numbers than this.)

Given an input n, it is possible to determine the number of numbers printed before and including the 1 is printed. For a given n this is called the *cycle-length* of n. In the example above, the cycle length of 22 is 16.

For any two numbers i and j you are to determine the maximum cycle length over all numbers between and including both i and j.

Input

The input will consist of a series of pairs of integers i and j, one pair of integers per line. All integers will be less than 10,000 and greater than 0.

You should process all pairs of integers and for each pair determine the maximum cycle length over all integers between and including i and j.

You can assume that no operation overflows a 32-bit integer.

Output

For each pair of input integers i and j you should output i, j, and the maximum cycle length for integers between and including i and j. These three numbers should be separated by at least one space with all three numbers on one line and with one line of output for each line of input. The integers i and j must appear in the output in the same order in which they appeared in the input and should be followed by the maximum cycle length (on the same line).

Sample Input

```
1 10
100 200
201 210
900 1000
```

Sample Output

```
1 10 20
100 200 125
201 210 89
900 1000 174
```

LC-Display

A friend of you has just bought a new computer. Until now, the most powerful computer he ever used has been a pocket calculator. Now, looking at his new computer, he is a bit disappointed, because he liked the LC-display of his calculator so much. So you decide to write a program that displays numbers in an LC-display-like style on his computer.

Input

The input file contains several lines, one for each number to be displayed. Each line contains two integers s, n ($1 \leq s \leq 10, 0 \leq n \leq 99\,999\,999$), where n is the number to be displayed and s is the size in which it shall be displayed.

The input file will be terminated by a line containing two zeros. This line should not be processed.

Output

Output the numbers given in the input file in an LC-display-style using s '-' signs for the horizontal segments and s '|' signs for the vertical ones. Each digit occupies exactly $s + 2$ columns and $2s + 3$ rows. (Be sure to fill all the white space occupied by the digits with blanks, also for the last digit.) There has to be exactly one column of blanks between two digits.

Output a blank line after each number. (You will find a sample of each digit in the sample output.)

Sample Input

```
2 12345
3 67890
0 0
```

Sample Output

```
   --   --        --
  |  |  |  |  |  |  |
  |  |  |  |  |  |  |
   --   --   --   --
  |  |     |  |     |
  |  |     |  |     |
   --   --        --


 ---   ---   ---   ---   ---
|         | |  | |  | |  | |
|         | |  | |  | |  | |
|         | |  | |  | |  | |
 ---           ---   ---
|   |     | |  |     | |  | |
|   |     | |  |     | |  | |
|   |     | |  |     | |  | |
 ---           ---   ---   ---
```

The Trip

A number of students are members of a club that travels annually to exotic locations. Their destinations in the past have included Indianapolis, Phoenix, Nashville, Philadelphia, San Jose, and Atlanta. This spring they are planning a trip to Eindhoven.

The group agrees in advance to share expenses equally, but it is not practical to have them share every expense as it occurs. So individuals in the group pay for particular things, like meals, hotels, taxi rides, plane tickets, etc. After the trip, each student's expenses are tallied and money is exchanged so that the net cost to each is the same, to within one cent. In the past, this money exchange has been tedious and time consuming. Your job is to compute, from a list of expenses, the minimum amount of money that must change hands in order to equalize (within a cent) all the students' costs.

Input

Standard input will contain the information for several trips. The information for each trip consists of a line containing a positive integer, n, the number of students on the trip, followed by n lines of input, each containing the amount, in dollars and cents, spent by a student. There are no more than 1000 students and no student spent more than $10,000.00. A single line containing 0 follows the information for the last trip.

Output

For each trip, output a line stating the total amount of money, in dollars and cents, that must be exchanged to equalize the students' costs.

Sample Input

```
3
10.00
20.00
30.00
4
15.00
15.01
3.00
3.01
0
```

Sample Output

```
$10.00
$11.99
```

Minesweeper

Have you ever played Minesweeper? It's a cute little game which comes within a certain Operating System which name we can't really remember. Well, the goal of the game is to find where are all the mines within a $M \times N$ field. To help you, the game shows a number in a square which tells you how many mines there are adjacent to that square. For instance, supose the following 4×4 field with 2 mines (which are represented by an '*' character):

```
*...
....
.*..
....
```

If we would represent the same field placing the hint numbers described above, we would end up with:

```
*100
2210
1*10
1110
```

As you may have already noticed, each square may have at most 8 adjacent squares.

Input

The input will consist of an arbitrary number of fields. The first line of each field contains two integers n and m ($0 < n, m \leq 100$) which stands for the number of lines and columns of the field respectively. The next n lines contains exactly m characters and represent the field.

Each safe square is represented by an '.' character (without the quotes) and each mine square is represented by an '*' character (also without the quotes). The first field line where $n = m = 0$ represents the end of input and should not be processed.

Output

For each field, you must print the following message in a line alone:

`Field #`x`:`

Where x stands for the number of the field (starting from 1). The next n lines should contain the field with the '.' characters replaced by the number of adjacent mines to that square. There must be an empty line between field outputs.

Sample Input

```
4 4
*...
....
.*..
....
3 5
**...
.....
.*...
0 0
```

Sample Output

```
Field #1:
*100
2210
1*10
1110

Field #2:
**100
33200
1*100
```

Jolly Jumpers

A sequence of $n > 0$ integers is called a *jolly jumper* if the absolute values of the difference between successive elements take on all the values 1 through $n - 1$. For instance,

```
1 4 2 3
```

is a jolly jumper, because the absolutes differences are 3, 2, and 1 respectively. The definition implies that any sequence of a single integer is a jolly jumper. You are to write a program to determine whether or not each of a number of sequences is a jolly jumper.

Input

Each line of input contains an integer $n \leq 3000$ followed by n integers representing the sequence.

Output

For each line of input, generate a line of output saying '`Jolly`' or '`Not jolly`'.

Sample Input

```
4 1 4 2 3
5 1 4 2 -1 6
```

Sample Output

```
Jolly
Not jolly
```

Hartals

A social research organization has determined a simple set of parameters to simulate the behavior of the political parties of our country. One of the parameters is a positive integer h (called the *hartal parameter*) that denotes the average number of days between two successive *hartals* (strikes) called by the corresponding party. Though the parameter is far too simple to be flawless, it can still be used to forecast the damages caused by *hartals*. The following example will give you a clear idea:

Consider three political parties. Assume $h_1 = 3$, $h_2 = 4$ and $h_3 = 8$ where h_i is the *hartal parameter* for party i ($i = 1, 2, 3$). Now, we will simulate the behavior of these three parties for $N = 14$ days. One must always start the simulation on a Sunday and assume that there will be no *hartals* on weekly holidays (on Fridays and Saturdays).

Days	1 Su	2 Mo	3 Tu	4 We	5 Th	6 Fr	7 Sa	8 Su	9 Mo	10 Tu	11 We	12 Th	13 Fr	14 Sa
Party 1			x			x			x			x		
Party 2				x				x				x		
Party 3								x						
Hartals			1	2				3	4			5		

The simulation above shows that there will be exactly 5 *hartals* (on days 3, 4, 8, 9 and 12) in 14 days. There will be no *hartal* on day 6 since it is a Friday. Hence we lose 5 working days in 2 weeks.

In this problem, given the hartal parameters for several political parties and the value of N, your job is to determine the number of working days we lose in those N days.

Input

The first line of the input consists of a single integer T giving the number of test cases to follow.

The first line of each test case contains an integer N ($7 \leq N \leq 3650$) giving the number of days over which the simulation must be run. The next line contains another integer P ($1 \leq P \leq 100$) representing the number of political parties in this case. The ith of the next P lines contains a positive integer h_i (which will never be a multiple of 7) giving the *hartal parameter* for party i ($1 \leq i \leq P$).

Output

For each test case in the input output the number of working days we lose. Each output must be on a separate line.

Sample Input

```
2
14
3
3
4
8
100
4
12
15
25
40
```

Sample Output

```
5
15
```

```
WERTYU
```

A common typing error is to place the hands on the keyboard one row to the right of the correct position. So 'Q' is typed as 'W' and 'J' is typed as 'K' and so on. You are to decode a message typed in this manner.

Input

Input consists of several lines of text. Each line may contain digits, spaces, upper case letters (except `Q`, `A`, `Z`), or punctuation shown above [except back-quote (`)]. Keys labelled with words [`Tab`, `BackSp`, `Control`, etc.] are not represented in the input.

Output

You are to replace each letter or punction symbol by the one immediately to its left on the 'QWERTY' keyboard shown above. Spaces in the input should be echoed in the output.

Sample Input

```
O S, GOMR YPFSU/
```

Sample Output

```
I AM FINE TODAY.
```

Common Permutation

Given two strings of lowercase letters, a and b, print the longest string x of lowercase letters such that there is a permutation of x that is a subsequence of a and there is a permutation of x that is a subsequence of b.

Input

Input file contains several lines of input. Consecutive two lines make a set of input. That means in the input file line 1 and 2 is a set of input, line 3 and 4 is a set of input and so on. The first line of a pair contains a and the second contains b. Each string is on a separate line and consists of at most 1000 lowercase letters.

Output

For each set of input, output a line containing x. If several x satisfy the criteria above, choose the first one in alphabetical order.

Sample Input

```
pretty
women
walking
down
the
street
```

Sample Output

```
e
nw
et
```

Vito's Family

The world-known gangster Vito Deadstone is moving to New York. He has a very big family there, all of them living in Lamafia Avenue. Since he will visit all his relatives very often, he is trying to find a house close to them.

Vito wants to minimize the total distance to all of them and has blackmailed you to write a program that solves his problem.

Input

The input consists of several test cases. The first line contains the number of test cases.

For each test case you will be given the integer number of relatives r ($0 < r < 500$) and the street numbers (also integers) $s_1, s_2, \ldots, s_i, \ldots, s_r$ where they live ($0 < s_i < 30000$). Note that several relatives could live in the same street number.

Output

For each test case your program must write the minimal sum of distances from the optimal Vito's house to each one of his relatives. The distance between two street numbers s_i and s_j is $d_{ij} = |s_i - s_j|$.

Sample Input

```
2
2 2 4
3 2 4 6
```

Sample Output

```
2
4
```

Shoemaker's Problem

Shoemaker has N jobs (orders from customers) which he must make. Shoemaker can work on only one job in each day. For each i-th job, it is known the integer T_i ($1 \leq T_i \leq 1000$), the time in days it takes the shoemaker to finish the job. For each day of delay before starting to work for the i-th job, shoemaker must pay a fine of S_i ($1 \leq S_i \leq 10000$) cents. Your task is to help the shoemaker, writing a programm to find the sequence of jobs with minimal total fine.

Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

First line of input contains an integer N ($1 \leq N \leq 1000$). The next N lines each contain two numbers: the time and fine of each task in order.

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

You programm should print the sequence of jobs with minimal fine. Each job should be represented by its number in input. All integers should be placed on only one output line and separated by one space. If multiple solutions are possible, print the first lexicographically.

Sample Input

```
1

4
3 4
1 1000
2 2
5 5
```

Sample Output

```
2 1 3 4
```

Bridge

n people wish to cross a bridge at night. A group of at most two people may cross at any time, and each group must have a flashlight. Only one flashlight is available among the n people, so some sort of shuttle arrangement must be arranged in order to return the flashlight so that more people may cross.

Each person has a different crossing speed; the speed of a group is determined by the speed of the slower member. Your job is to determine a strategy that gets all n people across the bridge in the minimum time.

Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

The first line of input contains n, followed by n lines giving the crossing times for each of the people. There are not more than 1000 people and nobody takes more than 100 seconds to cross the bridge.

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

The first line of output must contain the total number of seconds required for all n people to cross the bridge. The following lines give a strategy for achieving this time. Each line contains either one or two integers, indicating which person or people form the next group to cross. (Each person is indicated by the crossing time specified in the input. Although many people may have the same crossing time the ambiguity is of no consequence.)

Note that the crossings alternate directions, as it is necessary to return the flashlight so that more may cross. If more than one strategy yields the minimal time, any one will do.

Sample Input

```
1

4
1
2
5
10
```

Sample Output

```
17
1 2
1
5 10
2
1 2
```

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.
```

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.

```
      195   Initial number
      591
     ----
      786
      687
     ----
For example: 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 : $minimum_number_of_iterations($ and $the_resulting_palindrome_itself$ separated by one space.

Sample Input

```
3
195
265
750
```

Sample Output

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

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

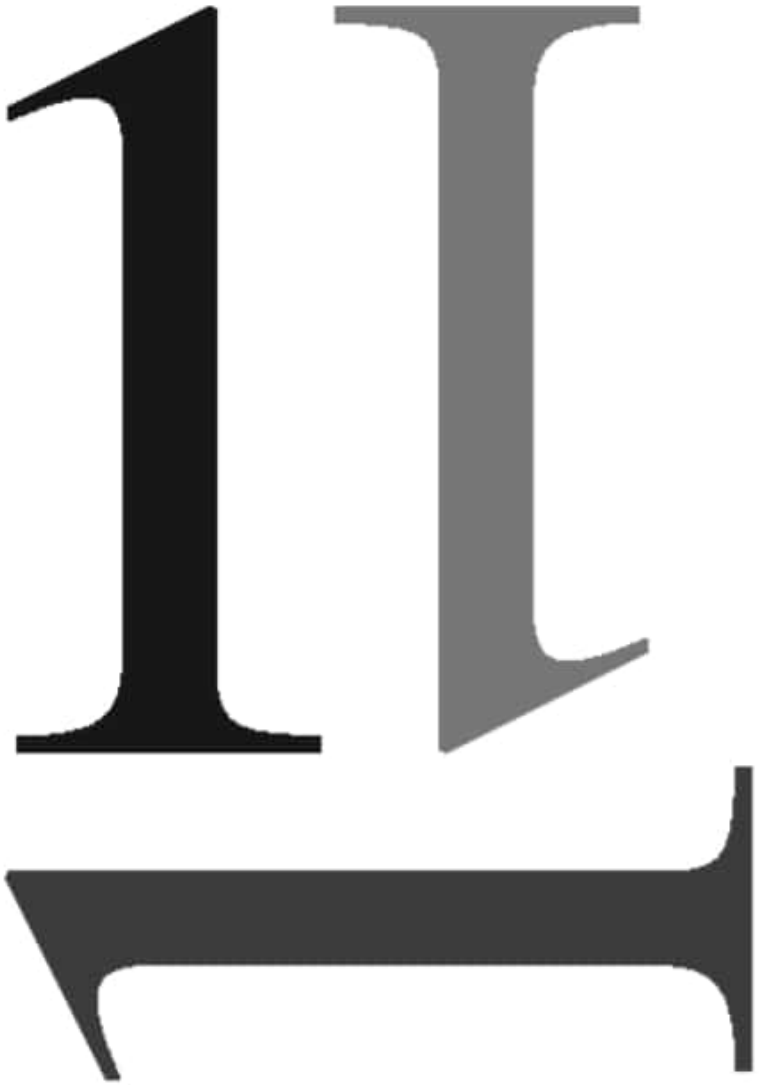
Each output line gives the smallest integer $x > 0$ such that $p = \sum_{i=0}^{x-1} 1 \times 10^i = a \times b$, where a is the corresponding input integer, and b is an integer greater than zero.

Sample Input

```
3
7
9901
```

Sample Output

```
3
6
12
```



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.
```

How Many Fibs

Recall the definition of the Fibonacci numbers:

$$\begin{aligned} f_1 &:= 1 \\ f_2 &:= 2 \\ f_n &:= f_{n-1} + f_{n-2} \qquad (n \geq 3) \end{aligned}$$

Given two numbers a and b, calculate how many Fibonacci numbers are in the range $[a, b]$.

Input

The input contains several test cases. Each test case consists of two non-negative integer numbers a and b. Input is terminated by $a = b = 0$. Otherwise, $a \leq b \leq 10^{100}$. The numbers a and b are given with no superfluous leading zeros.

Output

For each test case output on a single line the number of Fibonacci numbers f_i with $a \leq f_i \leq b$.

Sample Input

```
10 100
1234567890 9876543210
0 0
```

Sample Output

```
5
4
```

How Many Pieces of Land

You are given an elliptical shaped land and you are asked to choose n arbitrary points on its boundary. Then you connect all these points with one another with straight lines (that's $n * (n-1)/2$ connections for n points). What is the maximum number of pieces of land you will get by choosing the points on the boundary carefully?

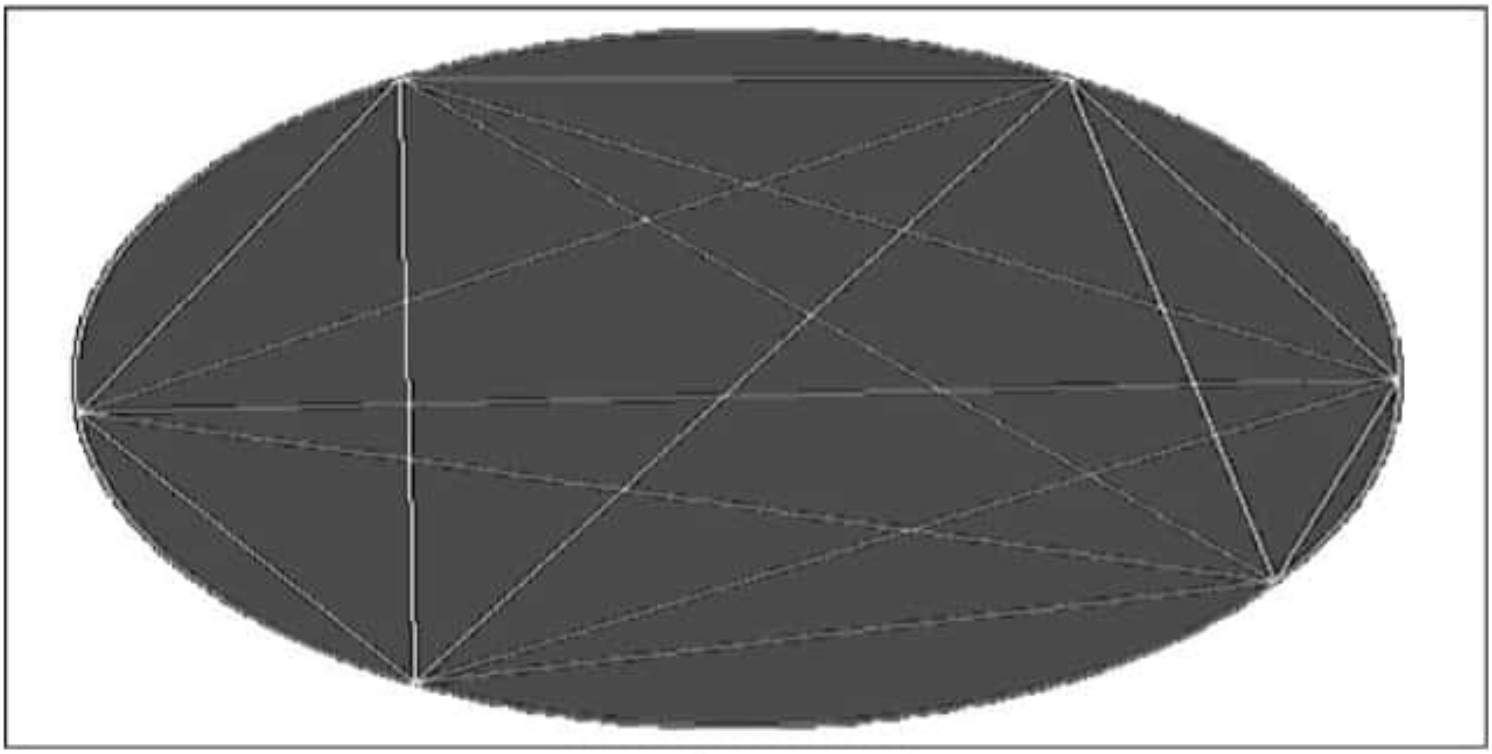


Fig: When the value of n is 6

Input

The first line of the input file contains one integer S ($0 < S < 3500$), which indicates how many sets of input are there. The next S lines contain S sets of input. Each input contains one integer N ($0 \leq N < 2^{31}$).

Output

For each set of input you should output in a single line the maximum number pieces of land possible to get for the value of N.

Sample Input

```
4
1
2
3
4
```

Sample Output

```
1
2
4
8
```

Counting

Gustavo knows how to count, but he is now learning how write numbers. As he is a very good student, he already learned 1, 2, 3 and 4. But he didn't realize yet that 4 is different than 1, so he thinks that 4 is another way to write 1. Besides that, he is having fun with a little game he created himself: he make numbers (with those four digits) and sum their values. For instance:

132 = 1 + 3 + 2 = 6
112314 = 1 + 1 + 2 + 3 + 1 + 1 = 9 (remember that Gustavo thinks that 4 = 1)

After making a lot of numbers in this way, Gustavo now wants to know how much numbers he can create such that their sum is a number n. For instance, for $n = 2$ he noticed that he can make 5 numbers: 11, 14, 41, 44 and 2 (he knows how to count them up, but he doesn't know how to write five). However, he can't figure it out for n greater than 2. So, he asked you to help him.

Input

Input will consist on an arbitrary number of sets. Each set will consist on an integer n such that $1 \leq n \leq 1000$. You must read until you reach the end of file.

Output

For each number read, you must output another number (on a line alone) stating how much numbers Gustavo can make such that the sum of their digits is equal to the given number.

Sample Input

```
2
3
```

Sample Output

```
5
13
```

Self-describing Sequence

Solomon Golomb's *self–describing sequence* $\langle f(1), f(2), f(3), \ldots \rangle$ is the only nondecreasing sequence of positive integers with the property that it contains exactly $f(k)$ occurrences of k for each k. A few moments thought reveals that the sequence must begin as follows:

n	1	2	3	4	5	6	7	8	9	10	11	12
$f(n)$	1	2	2	3	3	4	4	4	5	5	5	6

In this problem you are expected to write a program that calculates the value of $f(n)$ given the value of n.

Input

The input may contain multiple test cases. Each test case occupies a separate line and contains an integer n ($1 \leq n \leq 2,000,000,000$). The input terminates with a test case containing a value 0 for n and this case must not be processed.

Output

For each test case in the input output the value of $f(n)$ on a separate line.

Sample Input

```
100
9999
123456
1000000000
0
```

Sample Output

```
21
356
1684
438744
```