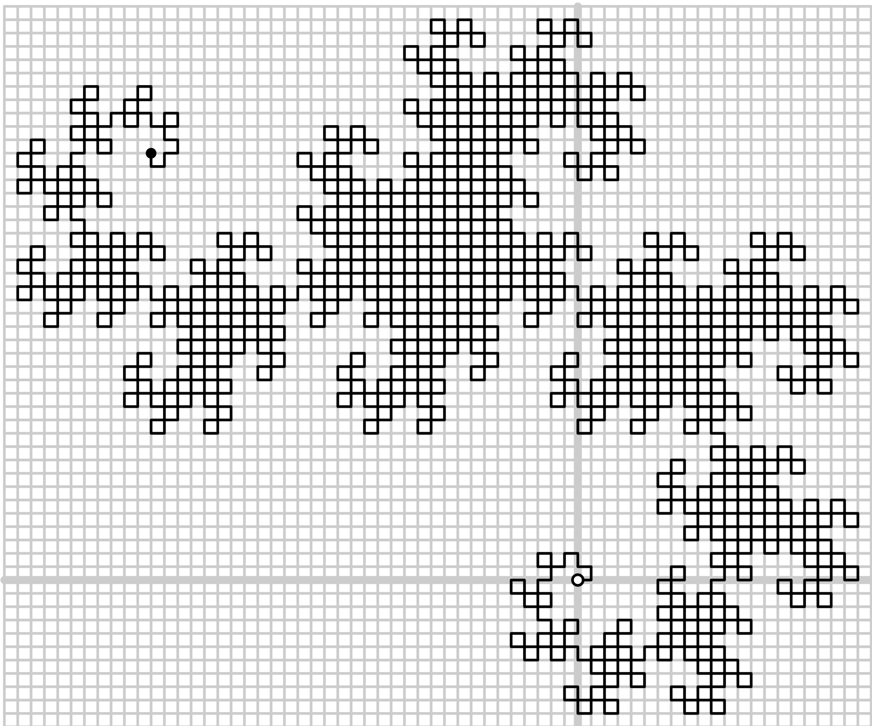
Problem A – Fractal

Define $r(s)$ to be the complement of the reverse of the binary string s. i.e. Reverse s and then convert all 1's to 0's and all 0's to 1's. Further define a sequence of binary string as follow: s_0 = `1` and $s_n = s_{n-1}1r(s_{n-1})$. i.e.

s_0 = `1`
s_1 = `1`**`1`**`0`
s_2 = `110`**`1`**`100`
s_3 = `1101100`**`1`**`1100100`
…

We then program a robot to move at a steady speed of 1 unit per second and make a right-angle turn according to the characters of s_{10} after every unit of movement. At the k^{th} turn, the robot turns to left if the k^{th} character of s_{10} is a 1, and to right otherwise. The figure below shows the whole path of the robot.



The robot is placed at the origin (the small circle) and face east originally. It ends up at the coordinates (-32,32) (the small spot) after 2048 seconds. The path of the robot is known as a dragon curve, a pretty well-known pattern of fractal.

If the robot is now programmed with input string s_{30} (with identical initial conditions as above), it will keep moving and then stop after 2^{31} seconds. We want to know the location of the robot at any given time.

Input

Input consists of multiple problem instances. Each instance consists of a single non-negative integer n, where $n \leq 10^9$. The input data is terminated by a negative integer. There will be less than 5000 test cases.

Output

For each input integer n, print out the location of the robot right after n second since the robot starts its journey with input string s_{30}. The location should be printed with the format "`(x,y)`" in a single line.

Sample Input

```
1
2
3
2048
1000000000
-1
```

Sample Output

```
(1,0)
(1,1)
(0,1)
(-32,32)
(9648,-31504)
```

Problem B – Consecutive Integers

Any positive integer can be written as the sum of several consecutive integers. For example,

$$15 = 1 + \ldots + 5 = 4 + \ldots + 6 = 7 + \ldots + 8 = 15 + \ldots + 15$$

Given a positive integer n, what are the consecutive positive integers with sum being n? If there are multiple solutions, which one consists of more numbers?

Input

Input consists of multiple problem instances. Each instance consists of a single positive integer n, where $n \leq 10^9$. The input data is terminated by a line containing -1. There will be at most 1000 test cases.

Output

For each input integer n, print out the desired solution with the format:

```
N = A + ... + B
```

in a single line.

(Read sample output for a clearer representation of the exact formatting.)

Sample Input

```
8
15
35
-1
```

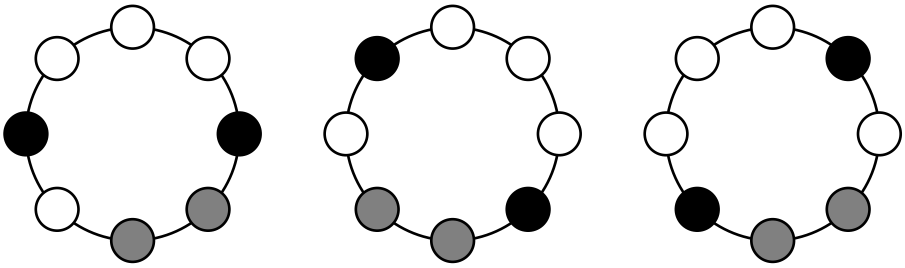
Sample Output

```
8 = 8 + ... + 8
15 = 1 + ... + 5
35 = 2 + ... + 8
```

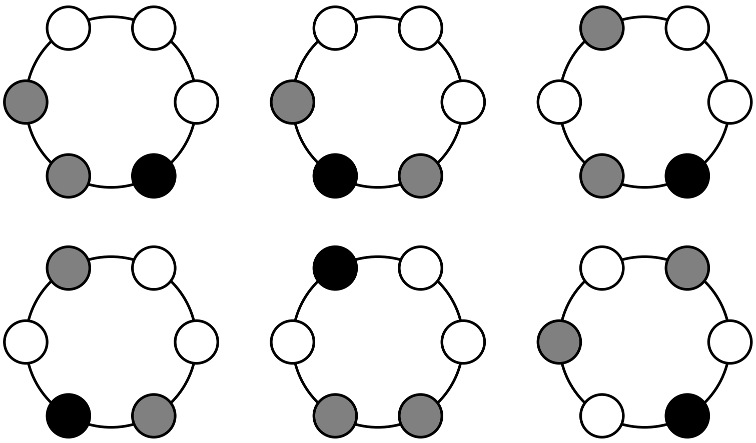
Problem C – Necklace

Once upon a time, three girls – Winnie, Grace and Bonnie – owned a large number of pearls. However, each of them only had a single color of pearls. Winnie had white pearls, Grace had grey pearls and Bonnie had black pearls. One day, after a long discussion, they decided to make necklaces using the pearls. They are interested in knowing how many patterns can be formed using a certain number of pearls of each color, and have asked you to solve this problem for them.

Note that rotating or flipping over a necklace cannot produce a different kind of necklace. i.e. The following figure shows three equivalent necklaces.



The following figure shows all possible necklaces formed by 3 white pearls, 2 grey pearls and 1 black pearl.



Input
The input begins with an integer *N* (≤ 2500) which indicates the number of test cases followed. Each of the following test cases consists of three non-negative integers *a*, *b*, *c*, where $3 \le a + b + c \le 40$.

Output
For each test case, print out the number of different necklaces that formed by *a* white pearls, *b* grey pearls and *c* black pearls in a single line.

Sample Input
```
2
3 2 1
2 2 2
```

Sample Output

```
6
11
```

Special thanks: Michael (for making up the story)

Problem D – Repetitive Multiple

An integer is said to be repetitive if it can be written as a concatenation of several copies (at least two) of another non-zero-leading integer. For example, 11, 123123, 454545 are all repetitive integers.

Since zero-leading integers are not allowed, 101 can not be considered as 0101. Therefore, 101 is not repetitive.

Given a positive integer *n*, what is the smallest repetitive integer which is a multiple of *n*.

Input
The input begins with an integer *N* (≤ 100) which indicates the number of test cases followed. Each of the following test cases consists of a positive integer *n*, where *n* will be less than 10^9.

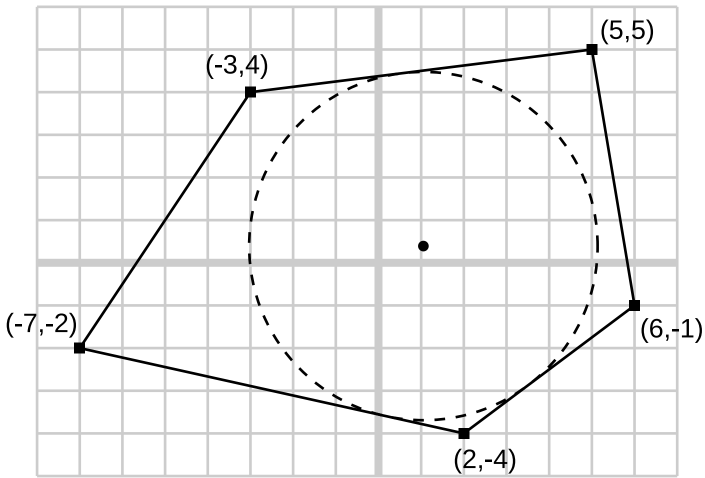
Output
For each test case, print out the smallest repetitive multiple of *n* in a single line.

Sample Input
```
5
7
101
123
999999
6339673
```

Sample Output
```
77
1010
33333
999999
114114114
```

Problem E – New Marketing Plan

You work at a company that produces circular plates. In a meeting, your boss announces a new marketing plan – the company will package the plates in specially shaped containers, so as to attract customers' attention. The specially shaped containers will always be convex polygon. The existing packaging software can easily compute the maximum plate size for a given container size, but unfortunately, it can only do so for rectangular containers. Therefore, your boss asks you to write a program that can compute the maximum plate size given the container's shape.



The above figure shows a maximum plate placed inside a convex-polygon-shaped container.

Given the shape of a container, you need to compute the radius of the largest possible plate which can be fitted inside the container?

Input

Input consists of multiple problem instances. Each instance begins with a single positive integer n, where $3 \le n \le 50$. It is followed by $2n$ integers, x_1, y_1, x_2, y_2, …, x_n, y_n, where $-1000 \le x_i, y_i \le 1000$ for all i. They indicate the shape of a container which is a simple convex polygon with n vertices (x_1, y_1), (x_2, y_2), …, (x_n, y_n) in counter-clockwise order. The area of the polygon is guaranteed to be positive and no three consecutive vertices are collinear. The input data is terminated by an n being 0. There will not be more than 100 test cases.

Output

For each test case, print out the radius of the largest possible plate which can be fitted inside the input container. The answer should be printed with 3 digits after the decimal point in a single line.

Sample Input

```
5
5 5 -3 4 -7 -2 2 -4 6 -1
3
0 0 10 0 0 10
0
```

Sample Output

```
4.082
2.929
```

Special thanks: Michael (for making up the story)

Problem F – String Partition

John was absurdly busy for preparing a programming contest recently. He wanted to create a ridiculously easy problem for the contest. His problem was not only easy, but also boring: Given a list of non-negative integers, what is the sum of them?

However, he made a very typical mistake when he wrote a program to generate the input data for his problem. He forgot to print out spaces to separate the list of integers. John quickly realized his mistake after looking at the generated input file because each line is simply a string of digits instead of a list of integers.

He then got a better idea to make his problem a little more interesting: There are many ways to split a string of digits into a list of non-zero-leading (0 itself is allowed) 32-bit *signed* integers. What is the maximum sum of the resultant integers if the string is split appropriately?

Input

The input begins with an integer N (≤ 500) which indicates the number of test cases followed. Each of the following test cases consists of a string of at most 200 digits.

Output

For each input, print out required answer in a single line.

Sample Input

```
6
1234554321
5432112345
000
121212121212
2147483648
11111111111111111111111111111111111111111111111111111
```

Sample Output

```
1234554321
543211239
0
2121212124
214748372
5555555666
```

Problem G – Coin Changing Again

There are four types of coins with value c_1, c_2, c_3 and c_4, and there are only d_1, d_2, d_3 and d_4 number of these coins respectively. How many ways are there to obtain a value v by adding up these coins?

For example, if you have 3 $1-coins, 2 $2-coins, 3 $5-coins, 1 $10-coin, there are 4 ways to obtain $10 from those coins:
10 = 1 + 1 + 1 + 2 + 5
10 = 1 + 2 + 2 + 5
10 = 5 + 5
10 = 10

Input
The input begins with an integer N ($\leq$ 100) which indicates the number of test cases followed. Each of the following test cases begins with five positive integers c_1, c_2, c_3, c_4, q, where $1 \leq c_1 < c_2 < c_3 < c_4 \leq 1000$ and $q \leq 100$. It is then followed by q queries. Each query consists of five integers, d_1, d_2, d_3, d_4, v, where $1 \leq d_1, d_2, d_3, d_4, v \leq 10^5$.

Output
For each query from each test case, print out the number of way to obtain v by adding up d_1 c_1-coins, d_2 c_2-coins, d_3 c_3-coins and d_4 c_4-coins in a single line.

Sample Input
```
2
1 2 5 10 2
3 2 3 1 10
1000 2 2 2 900
10 20 30 40 1
100 100 100 100 101
```

Sample Output
```
4
27
0
```