GPC-UPC Number Theory I Contest

A. From Hero to Zero

1 second, 256 megabytes

You are given an integer n and an integer k.

In one step you can do one of the following moves:

- decrease n by 1;
- divide n by k if n is divisible by k.

For example, if $n = 27$ and $k = 3$ you can do the following steps: $27 \rightarrow 26 \rightarrow 25 \rightarrow 24 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 2 \rightarrow 1 \rightarrow 0$.

You are asked to calculate the minimum number of steps to reach 0 from n.

Input

The first line contains one integer t ($1 \le t \le 100$) — the number of queries.

The only line of each query contains two integers n and k ($1 \le n \le 10^{18}$, $2 \le k \le 10^{18}$).

Output

For each query print the minimum number of steps to reach 0 from n in single line.

input

```
2
59 3
1000000000000000000 10
```

output

```
8
19
```

Steps for the first test case are: $59 \rightarrow 58 \rightarrow 57 \rightarrow 19 \rightarrow 18 \rightarrow 6 \rightarrow 2 \rightarrow 1 \rightarrow 0$.

In the second test case you have to divide n by k 18 times and then decrease n by 1.

B. Common Divisors

2 seconds, 256 megabytes

You are given an array a consisting of n integers.

Your task is to say the number of such positive integers x such that x divides **each** number from the array. In other words, you have to find the number of common divisors of all elements in the array.

For example, if the array a will be $[2, 4, 6, 2, 10]$, then 1 and 2 divide each number from the array (so the answer for this test is 2).

Input

The first line of the input contains one integer n ($1 \le n \le 4 \cdot 10^5$) — the number of elements in a.

The second line of the input contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^{12}$), where a_i is the i-th element of a.

Output

Print one integer — the number of such positive integers x such that x divides **each** number from the given array (in other words, the answer is the number of common divisors of all elements in the array).

input

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

output

```
1
```

input

```
6
6 90 12 18 30 18
```

output

```
4
```

C. Merge it!

1 second, 256 megabytes

You are given an array a consisting of n integers $a_1, a_2, \ldots, a_n$.

In one operation you can choose two elements of the array and replace them with the element equal to their sum (it does not matter where you insert the new element). For example, from the array $[2, 1, 4]$ you can obtain the following arrays: $[3, 4]$, $[1, 6]$ and $[2, 5]$.

Your task is to find the maximum possible number of elements divisible by 3 that are in the array after performing this operation an arbitrary (possibly, zero) number of times.

You have to answer t independent queries.

Input

The first line contains one integer t ($1 \le t \le 1000$) — the number of queries.

The first line of each query contains one integer n ($1 \le n \le 100$).

The second line of each query contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

Output

For each query print one integer in a single line — the maximum possible number of elements divisible by 3 that are in the array after performing described operation an arbitrary (possibly, zero) number of times.

input

```
2
5
3 1 2 3 1
7
1 1 1 1 1 2 2
```

output

```
3
3
```

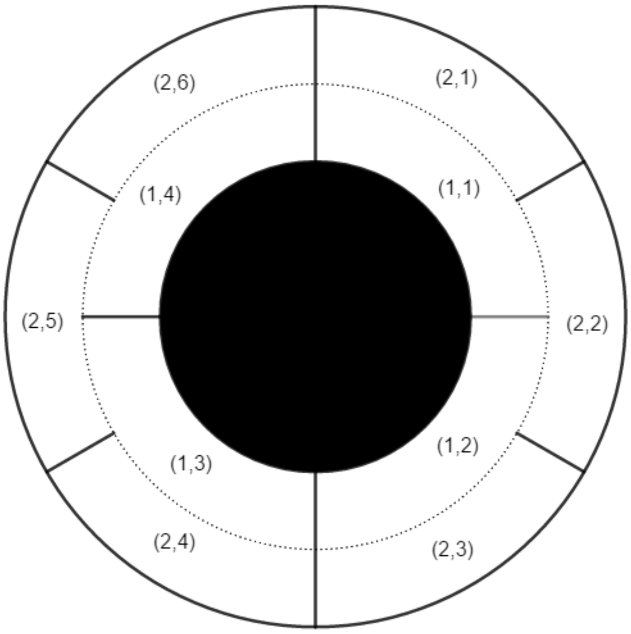
In the first query of the example you can apply the following sequence of operations to obtain 3 elements divisible by 3: $[3, 1, 2, 3, 1] \rightarrow [3, 3, 3, 1]$.

In the second query you can obtain 3 elements divisible by 3 with the following sequence of operations: $[1, 1, 1, 1, 1, 2, 2] \rightarrow [1, 1, 1, 1, 2, 3] \rightarrow [1, 1, 1, 3, 3] \rightarrow [2, 1, 3, 3] \rightarrow$
.

D. Round Corridor

1 second, 256 megabytes

Amugae is in a very large round corridor. The corridor consists of two areas. The inner area is equally divided by n sectors, and the outer area is equally divided by m sectors. A wall exists between each pair of sectors of same area (inner or outer), but there is no wall between the inner area and the outer area. A wall always exists at the 12 o'clock position.



The inner area's sectors are denoted as $(1,1), (1,2), \dots, (1,n)$ in clockwise direction. The outer area's sectors are denoted as $(2,1), (2,2), \dots, (2,m)$ in the same manner. For a clear understanding, see the example image above.

Amugae wants to know if he can move from one sector to another sector. He has q questions.

For each question, check if he can move between two given sectors.

Input

The first line contains three integers n, m and q ($1 \le n, m \le 10^{18}$, $1 \le q \le 10^4$) — the number of sectors in the inner area, the number of sectors in the outer area and the number of questions.

Each of the next q lines contains four integers s_x, s_y, e_x, e_y ($1 \le s_x, e_x \le 2$; if $s_x = 1$, then $1 \le s_y \le n$, otherwise $1 \le s_y \le m$; constraints on e_y are similar). Amague wants to know if it is possible to move from sector (s_x, s_y) to sector (e_x, e_y).

Output

For each question, print "YES" if Amugae can move from (s_x, s_y) to (e_x, e_y), and "NO" otherwise.

You can print each letter in any case (upper or lower).

input

```
4 6 3
1 1 2 3
2 6 1 2
2 6 2 4
```

output

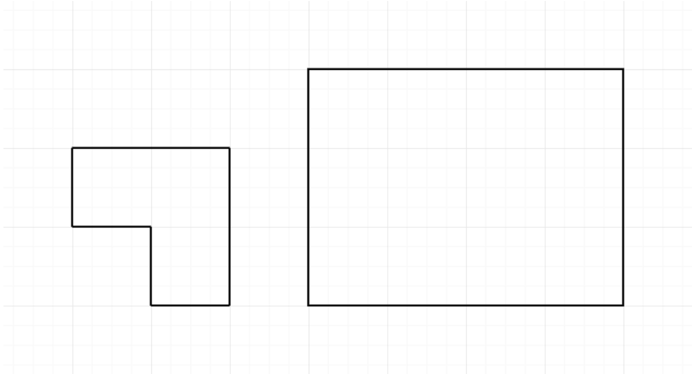
```
YES
NO
YES
```

Example is shown on the picture in the statement.

E. Filling Shapes

1 second, 256 megabytes

You have a given integer n. Find the number of ways to fill all $3 \times n$ tiles with the shape described in the picture below. Upon filling, no empty spaces are allowed. Shapes cannot overlap.



This picture describes when $n = 4$. The left one is the shape and the right one is $3 \times n$ tiles.

Input

The only line contains one integer n ($1 \le n \le 60$) — the length.

Output

Print the number of ways to fill.

input

```
4
```

output

```
4
```

input

```
1
```

output

```
0
```

In the first example, there are 4 possible cases of filling.

In the second example, you cannot fill the shapes in 3×1 tiles.

F. Ropewalkers

1 second, 256 megabytes

Polycarp decided to relax on his weekend and visited to the performance of famous ropewalkers: Agafon, Boniface and Konrad.

The rope is straight and infinite in both directions. At the beginning of the performance, Agafon, Boniface and Konrad are located in positions a, b and c respectively. At the end of the performance, the distance between each pair of ropewalkers was **at least** d.

Ropewalkers can walk on the rope. In one second, only one ropewalker can change his position. Every ropewalker can change his position exactly by 1 (i. e. shift by 1 to the left or right direction on the rope). Agafon, Boniface and Konrad **can not** move at the same time (**Only one of them can move at each moment**). Ropewalkers can be at the same positions at the same time and can "walk past each other".

You should find the minimum duration (in seconds) of the performance. In other words, find the minimum number of seconds needed so that the distance between each pair of ropewalkers can be greater or equal to d.

Ropewalkers can walk to negative coordinates, due to the rope is infinite to both sides.

Input

The only line of the input contains four integers a, b, c, d ($1 \le a, b, c, d \le 10^9$). It is possible that any two (or all three) ropewalkers are in the same position at the beginning of the performance.

Output

Output one integer — the minimum duration (in seconds) of the performance.

input

```
5 2 6 3
```

output

```
2
```

input

```
3 1 5 6
```

output

```
8
```

input

```
8 3 3 2
```

output

```
2
```

input

```
2 3 10 4
```

output

```
3
```

In the first example: in the first two seconds Konrad moves for 2 positions to the right (to the position 8), while Agafon and Boniface stay at their positions. Thus, the distance between Agafon and Boniface will be $|5-2|=3$, the distance between Boniface and Konrad will be $|2-8|=6$ and the distance between Agafon and Konrad will be $|5-8|=3$. Therefore, all three pairwise distances will be at least $d=3$, so the performance could be finished within 2 seconds.

G. Equalize Prices

1 second, 256 megabytes

There are n products in the shop. The price of the i-th product is a_i. The owner of the shop wants to equalize the prices of all products. However, he wants to change prices smoothly.

In fact, the owner of the shop can change the price of some product i in such a way that the difference between the old price of this product a_i and the new price b_i is at most k. In other words, the condition $|a_i - b_i| \le k$ should be satisfied ($|x|$ is the absolute value of x).

He can change the price for each product **not more than once**. Note that he can leave the old prices for some products. The new price b_i of each product i should be positive (i.e. $b_i > 0$ should be satisfied for all i from 1 to n).

Your task is to find out the **maximum** possible **equal** price B of **all** productts with the restriction that for all products the condiion $|a_i - B| \le k$ should be satisfied (where a_i is the old price of the product and B is the same new price of all products) or report that it is impossible to find such price B.

Note that the chosen price B should be integer.

You should answer q independent queries.

Input

The first line of the input contains one integer q ($1 \le q \le 100$) — the number of queries. Each query is presented by two lines.

The first line of the query contains two integers n and k ($1 \le n \le 100, 1 \le k \le 10^8$) — the number of products and the value k. The second line of the query contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 10^8$), where a_i is the price of the i-th product.

Output

Print q integers, where the i-th integer is the answer B on the i-th query.

If it is impossible to equalize prices of **all** given products with restriction that for all products the condition $|a_i - B| \le k$ should be satisfied (where a_i is the old price of the product and B is the new equal price of all products), print `-1`. Otherwise print the **maximum** possible equal price of **all** products.

input

```
4
5 1
1 1 2 3 1
4 2
6 4 8 5
2 2
1 6
3 5
5 2 5
```

output

```
2
6
-1
7
```

In the first example query you can choose the price $B=2$. It is easy to see that the difference between each old price and each new price $B=2$ is no more than 1.

In the second example query you can choose the price $B=6$ and then all the differences between old and new price $B=6$ will be no more than 2.

In the third example query you cannot choose any suitable price B. For any value B at least one condition out of two will be violated: $|1-B| \le 2$, $|6-B| \le 2$.

In the fourth example query all values B between 1 and 7 are valid. But the maximum is 7, so it's the answer.

H. Stickers and Toys

2 seconds, 256 megabytes

Your favorite shop sells n Kinder Surprise chocolate eggs. You know that exactly s stickers and exactly t toys are placed in n eggs in total.

Each Kinder Surprise can be one of three types:

- it can contain a single sticker and **no toy**;
- it can contain a single toy and **no sticker**;
- it can contain both a single sticker **and** a single toy.

But you **don't know** which type a particular Kinder Surprise has. All eggs look identical and indistinguishable from each other.

What is the minimum number of Kinder Surprise Eggs you have to buy to be sure that, whichever types they are, you'll obtain at least one sticker and at least one toy?

Note that you do not open the eggs in the purchasing process, that is, you just buy some number of eggs. It's guaranteed that the answer always exists.

Input

The first line contains the single integer T ($1 \le T \le 100$) — the number of queries.

Next T lines contain three integers n, s and t each ($1 \le n \le 10^9$, $1 \le s, t \le n$, $s + t \ge n$) — the number of eggs, stickers and toys.

All queries are independent.

Output

Print T integers (one number per query) — the minimum number of Kinder Surprise Eggs you have to buy to be sure that, whichever types they are, you'll obtain at least one sticker and one toy

input

```
3
10 5 7
10 10 10
2 1 1
```

output

```
6
1
2
```

In the first query, we have to take at least 6 eggs because there are 5 eggs with *only toy* inside and, in the worst case, we'll buy all of them.

In the second query, all eggs have both a sticker and a toy inside, that's why it's enough to buy only one egg.

In the third query, we have to buy both eggs: one with a sticker and one with a toy.

I. Sport Mafia

2 seconds, 256 megabytes

Each evening after the dinner the SIS's students gather together to play the game of Sport Mafia.

For the tournament, Alya puts candies into the box, which will serve as a prize for a winner. To do that, she performs n actions. The first action performed is to put a single candy into the box. For each of the remaining moves she can choose from two options:

- the first option, in case the box contains at least one candy, is to take **exactly one candy out and eat it**. This way the number of candies in the box decreased by 1;
- the second option is to put candies in the box. In this case, Alya will put 1 more candy, than she put in the previous time.

Thus, if the box is empty, then it can only use the second option.

For example, one possible sequence of Alya's actions look as follows:

- put one candy into the box;
- put two candies into the box;
- eat one candy from the box;
- eat one candy from the box;
- put three candies into the box;
- eat one candy from the box;
- put four candies into the box;
- eat one candy from the box;
- put five candies into the box;

This way she will perform 9 actions, the number of candies at the end will be 11, while Alya will eat 4 candies in total.

You know the total number of actions n and the number of candies at the end k. You need to find the total number of sweets Alya ate. That is the number of moves of the first option. It's guaranteed, that for the given n and k the answer always exists.

Please note, that during an action of the first option, Alya takes out and eats exactly one candy.

Input

The first line contains two integers n and k ($1 \le n \le 10^9$; $0 \le k \le 10^9$) — the total number of moves and the number of candies in the box at the end.

It's guaranteed, that for the given n and k the answer exists.

Output

Print a single integer — the number of candies, which Alya ate. Please note, that in this problem there aren't multiple possible answers — the answer is unique for any input data.

input
```
1 1
```
output
```
0
```

input
```
9 11
```
output
```
4
```

input
```
5 0
```
output
```
3
```

input
```
3 2
```
output
```
1
```

In the first example, Alya has made one move only. According to the statement, the first move is always putting one candy in the box. Hence Alya ate 0 candies.

In the second example the possible sequence of Alya's actions looks as follows:

- put 1 candy,
- put 2 candies,
- eat a candy,
- eat a candy,
- put 3 candies,
- eat a candy,
- put 4 candies,
- eat a candy,
- put 5 candies.

This way, she will make exactly $n = 9$ actions and in the end the box will contain $1+2-1-1+3-1+4-1+5=11$ candies. The answer is 4, since she ate 4 candies in total.

J. Count Pairs

4 seconds, 256 megabytes

You are given a **prime** number p, n integers $a_1, a_2, \ldots, a_n$, and an integer k.

Find the number of pairs of indexes (i, j) ($1 \le i < j \le n$) for which $(a_i + a_j)(a_i^2 + a_j^2) \equiv k \bmod p$.

Input

The first line contains integers n, p, k ($2 \le n \le 3 \cdot 10^5$, $2 \le p \le 10^9$, $0 \le k \le p-1$). p is guaranteed to be prime.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le p-1$). It is guaranteed that all elements are different.

Output

Output a single integer — answer to the problem.

input
```
3 3 0
0 1 2
```
output
```
1
```

input
```
6 7 2
1 2 3 4 5 6
```
output
```
3
```

In the first example:

$(0+1)(0^2+1^2) = 1 \equiv 1 \bmod 3$.

$(0+2)(0^2+2^2) = 8 \equiv 2 \bmod 3$.

$(1+2)(1^2+2^2) = 15 \equiv 0 \bmod 3$.

So only 1 pair satisfies the condition.

In the second example, there are 3 such pairs: $(1, 5)$, $(2, 3)$, $(4, 6)$.

K. Fractions

1 second, 1024 megabytes

About 44 days are left before *College Scholastic Ability Test* is held. This exam aims to measure students' achievement of National Curriculum standards and scholastic ability required for college education. (http://www.kice.re.kr/sub/info.do?m=0205&s=english)

One of the subjects covered by this test is Mathematics, which consists of 21 multiple choice questions and 9 short-answer questions. The answer of each short-answer question is guaranteed to be a *unique positive integer below 1 000*, as you can see from the answer sheet below.

However, the organizers might want to give students short-answer questions with non-integer answers, such as $2\sqrt{3}$ or $\frac{5}{3}$. Usually, the workaround is to write the answer in a canonical form, and then sum up all the integers inside that form and ask students to write that number instead.

In particular, when the answer is a positive rational number $\frac{a}{b}$, the organizers usually ask students to reduce it and sum up the numerator and the denominator of the reduced fraction. For example, when the answer is $\frac{18}{10}$, the student should reduce it to $\frac{9}{5}$ and write the final answer as $9 + 5 = 14$.

However, when the answer is $\frac{521}{500}$, the reduced fraction is also $\frac{521}{500}$, so the student should write the final answer as $521 + 500 = 1021$. But this shouldn't happen, since all the answers for the short-answer questions are below 1 000. To avoid this situation, the organizers should make sure that after reducing the fraction, the sum of the numerator and the denominator shouldn't exceed 999. Let's call such fractions as *Suneung Fractions*. For example, $\frac{1996}{2}$ and $\frac{18}{10}$ are *Suneung fractions*, while $\frac{1998}{2}$ and $\frac{521}{500}$ are not.

Suppose that, this year, one of the organizers wrote a problem, and the answer to that problem is $\frac{x}{y}$. Since the problem is not finalized yet, the only thing we know is $A \le x \le B$ and $C \le y \le D$ holds, for given A, B, C, D. The organizers want to know, among all the pairs (x, y), how many of $\frac{x}{y}$ is a *Suneung fraction*. Write a program that counts this number.

Input

The first and only line contains four space-separated integers A, B, C and D ($1 \le A \le B \le 10^{12}, 1 \le C \le D \le 10^{12}$)

Output

Print the number of integral pairs (x, y) ($A \le x \le B, C \le y \le D$), where $\frac{x}{y}$ is a *Suneung fraction*.

input
```
5 8 3 6
```
output
```
16
```

input
```
2018 2019 2018 2019
```
output
```
2
```

L. Multiplication Dilemma

1 second, 256 megabytes

Multiplication operation is not always easy! For example, it is hard to calculate 27×20 using your mind, but it is easier to find the answer using the following methods: $30 \times 20 - 3 \times 20$. It turns out that people can calculate the multiplication of two special numbers very easily.

A number is called special if it contains exactly one non-zero digit at the beginning (i.e. the most significant digit), followed by a non-negative number of zeros. For example, 30, 7, 5000 are special numbers, while 0, 11, 8070 are not.

In this problem, you are given two numbers a and b. Your task is to calculate the multiplication of a and b ($a \times b$), by writing the multiplication expression as a summation or subtraction of multiplication of special numbers. Can you?

Input

The first line contains an integer T ($1 \le T \le 10^4$) specifying the number of test cases.

Each test case consists of a single line containing two integers a and b ($-10^9 \le a, b \le 10^9$, $a \ne 0$, $b \ne 0$), as described in the problem statement above.

Output

For each test case, print a single line containing the multiplication expression of a and b as a summation or subtraction of multiplication of special numbers. All special numbers must be between -10^9 and 10^9 (inclusive). If there are multiple solutions, print any of them. It is guaranteed that an answer always exists for the given input.

The multiplication expression must be printed in exactly one line. A single term must be printed as in which z and w are both special numbers, # represents a single space, and x represents the multiplication operation. Two consecutive terms must be printed as in which z and w are both special numbers, # represents a single space, represents the multiplication operation, and represents either the addition operation $+$ or the subtraction operation $-$. (**Check the sample output for more clarification**).

input
```
2
55 20
70 17
```
output
```
60 x 20 - 5 x 20
-3 x 70 + 70 x 20
```

M. Weird Requirements

1.5 seconds, 256 megabytes

It is hard to find teams for students to participate in ACM contests, most students have weird requirements for their team mates. For example, Ziad wants the GCD of all his team mate's ratings on all famous websites to be exactly X and the LCM of his ratings to be exactly Y.

There are N famous competitive programming websites. Ramzi's rating on the i^{th} website is A_i. Ramzi is very experienced, in one contest, he can change his rating on one website to any other value. What is the minimum number of contests that Ramzi must participate in, so that Ziad accepts him as his team mate?

Input

The first line contains one integer T, the number of test cases.

Each test case starts with one line containing three space-separated integers N, X, and Y, the number of famous websites, the required GCD value, and the required LCM value respectively. $(1 \le N \le 10^5)$ $(1 \le X, Y \le 10^9)$

The next line contains N space-separated integers A_i, Ramzi's rating on all websites. $(1 \le A_i \le 10^9)$

Output

For each test case print one line containing one integer, the minimum number of contests Ramzi must participate in so that he can become Ziad's team mate.

If it is impossible for Ziad to accept him as his team mate , print -1.

input
```
1
5 1 10
1 5 1 10 10
```
output
```
0
```

The Greatest Common Divisor (GCD) of an array is the maximum number that divides all integers in the array without remainder.

The Least Common Multiple (LCM) of an array is the minimum positive number that is a multiple of all integers in the array.

N. Congruence Equation

3 seconds, 256 megabytes

Given an integer x. Your task is to find out how many positive integers n ($1 \le n \le x$) satisfy

$$n \cdot a^n \equiv b \pmod p,$$

where a, b, p are all known constants.

Input

The only line contains four integers a, b, p, x ($2 \le p \le 10^6 + 3$, $1 \le a, b < p$, $1 \le x \le 10^{12}$). It is guaranteed that p is a prime.

Output

Print a single integer: the number of possible answers n.

input
```
2 3 5 8
```
output
```
2
```

input
```
4 6 7 13
```
output
```
1
```

input
```
233 233 10007 1
```
output
```
1
```

In the first sample, we can see that $n = 2$ and $n = 8$ are possible answers.

O. Martian Clock

2 seconds, 256 megabytes

Having stayed home alone, Petya decided to watch forbidden films on the Net in secret. "What ungentlemanly behavior!" — you can say that, of course, but don't be too harsh on the kid. In his country films about the Martians and other extraterrestrial civilizations are forbidden. It was very unfair to Petya as he adored adventure stories that featured lasers and robots.

Today Petya is watching a shocking blockbuster about the Martians called "R2:D2". What can "R2:D2" possibly mean? It might be the Martian time represented in the Martian numeral system. Petya knows that time on Mars is counted just like on the Earth (that is, there are 24 hours and each hour has 60 minutes). The time is written as "$a:b$", where the string a stands for the number of hours (from 0 to 23 inclusive), and string b stands for the number of minutes (from 0 to 59 inclusive). The only thing Petya doesn't know is in what numeral system the Martian time is written.

Your task is to print the radixes of all numeral system which can contain the time "$a:b$".

Input

The first line contains a single string as "$a:b$" (without the quotes). There a is a non-empty string, consisting of numbers and uppercase Latin letters. String a shows the number of hours. String b is a non-empty string that consists of numbers and uppercase Latin letters. String b shows the number of minutes. The lengths of strings a and b are from 1 to 5 characters, inclusive. Please note that strings a and b can have leading zeroes that do not influence the result in any way (for example, string "`008:1`" in decimal notation denotes correctly written time).

We consider characters `0`, `1`, ..., `9` as denoting the corresponding digits of the number's representation in some numeral system, and characters A, B, ..., Z correspond to numbers `10`, `11`, ..., `35`.

Output

Print the radixes of the numeral systems that can represent the time "$a:b$" in the increasing order. Separate the numbers with spaces or line breaks. If there is no numeral system that can represent time "$a:b$", print the single integer `0`. If there are infinitely many numeral systems that can represent the time "$a:b$", print the single integer `-1`.

Note that on Mars any positional numeral systems with positive radix strictly larger than one are possible.

input
```
11:20
```
output
```
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
```

input
```
2A:13
```
output
```
0
```

input
```
000B:00001
```
output
```
-1
```

Let's consider the first sample. String "`11:20`" can be perceived, for example, as time `4:6`, represented in the ternary numeral system or as time `17:32` in hexadecimal system.

Let's consider the second sample test. String "`2A:13`" can't be perceived as correct time in any notation. For example, let's take the base-11 numeral notation. There the given string represents time `32:14` that isn't a correct time.

Let's consider the third sample. String "`000B:00001`" can be perceived as a correct time in the infinite number of numeral systems. If you need an example, you can take any numeral system with radix no less than `12`.

P. Escape

2 seconds, 256 megabytes

The princess is going to escape the dragon's cave, and she needs to plan it carefully.

The princess runs at v_p miles per hour, and the dragon flies at v_d miles per hour. The dragon will discover the escape after t hours and will chase the princess immediately. Looks like there's no chance to success, but the princess noticed that the dragon is very greedy and not too smart. To delay him, the princess decides to borrow a couple of bijous from his treasury. Once the dragon overtakes the princess, she will drop one bijou to distract him. In this case he will stop, pick up the item, return to the cave and spend f hours to straighten the things out in the treasury. Only after this will he resume the chase again from the very beginning.

The princess is going to run on the straight. The distance between the cave and the king's castle she's aiming for is c miles. How many bijous will she need to take from the treasury to be able to reach the castle? If the dragon overtakes the princess at exactly the same moment she has reached the castle, we assume that she reached the castle before the dragon reached her, and doesn't need an extra bijou to hold him off.

Input
The input data contains integers v_p, v_d, t, f and c, one per line ($1 \le v_p, v_d \le 100$, $1 \le t, f \le 10$, $1 \le c \le 1000$).

Output
Output the minimal number of bijous required for the escape to succeed.

input
```
1
2
1
1
10
```
output
```
2
```

input
```
1
2
1
1
8
```
output
```
1
```

In the first case one hour after the escape the dragon will discover it, and the princess will be 1 mile away from the cave. In two hours the dragon will overtake the princess 2 miles away from the cave, and she will need to drop the first bijou. Return to the cave and fixing the treasury will take the dragon two more hours; meanwhile the princess will be 4 miles away from the cave. Next time the dragon will overtake the princess 8 miles away from the cave, and she will need the second bijou, but after this she will reach the castle without any further trouble.

The second case is similar to the first one, but the second time the dragon overtakes the princess when she has reached the castle, and she won't need the second bijou.

Q. Insomnia cure

2 seconds, 256 megabytes

«One dragon. Two dragon. Three dragon», — the princess was counting. She had trouble falling asleep, and she got bored of counting lambs when she was nine.

However, just counting dragons was boring as well, so she entertained herself at best she could. Tonight she imagined that all dragons were here to steal her, and she was fighting them off. Every k-th dragon got punched in the face with a frying pan. Every l-th dragon got his tail shut into the balcony door. Every m-th dragon got his paws trampled with sharp heels. Finally, she threatened every n-th dragon to call her mom, and he withdrew in panic.

How many imaginary dragons suffered moral or physical damage tonight, if the princess counted a total of d dragons?

Input
Input data contains integer numbers k, l, m, n and d, each number in a separate line ($1 \le k, l, m, n \le 10$, $1 \le d \le 10^5$).

Output
Output the number of damaged dragons.

input
```
1
2
3
4
12
```
output
```
12
```

input
```
2
3
4
5
24
```
output
```
17
```

In the first case every first dragon got punched with a frying pan. Some of the dragons suffered from other reasons as well, but the pan alone would be enough.

In the second case dragons 1, 7, 11, 13, 17, 19 and 23 escaped unharmed.

R. Three Garlands

2 seconds, 256 megabytes

Mishka is decorating the Christmas tree. He has got three garlands, and all of them will be put on the tree. After that Mishka will switch these garlands on.

When a garland is switched on, it periodically changes its state — sometimes it is lit, sometimes not. Formally, if i-th garland is switched on during x-th second, then it is lit only during seconds x, $x + k_i$, $x + 2k_i$, $x + 3k_i$ and so on.

Mishka wants to switch on the garlands in such a way that during each second after switching the garlands on there would be at least one lit garland. Formally, Mishka wants to choose three integers x_1, x_2 and x_3 (not necessarily distinct) so that he will switch on the first garland during x_1-th second, the second one — during x_2-th second, and the third one — during x_3-th second, respectively, and during each second starting from $max(x_1, x_2, x_3)$ at least one garland will be lit.

Help Mishka by telling him if it is possible to do this!

Input
The first line contains three integers k_1, k_2 and k_3 ($1 \le k_i \le 1500$) — time intervals of the garlands.

Output
If Mishka can choose moments of time to switch on the garlands in such a way that each second after switching the garlands on at least one garland will be lit, print YES.

Otherwise, print NO.

input
```
2 2 3
```
output
```
YES
```

input
```
4 2 3
```
output
```
NO
```

In the first example Mishka can choose $x_1 = 1$, $x_2 = 2$, $x_3 = 1$. The first garland will be lit during seconds $1, 3, 5, 7, ...$, the second — $2, 4, 6, 8, ...$, which already cover all the seconds after the 2-nd one. It doesn't even matter what x_3 is chosen. Our choice will lead third to be lit during seconds $1, 4, 7, 10, ...$, though.

In the second example there is no way to choose such moments of time, there always be some seconds when no garland is lit.

S. Unary

2 seconds, 256 megabytes

Unary is a minimalistic Brainfuck dialect in which programs are written using only one token.

Brainfuck programs use 8 commands: "+", "-", "[", "]", "<", ">", "." and "," (their meaning is not important for the purposes of this problem). Unary programs are created from Brainfuck programs using the following algorithm. First, replace each command with a corresponding binary code, using the following conversion table:

- ">" → 1000,
- "<" → 1001,
- "+" → 1010,
- "-" → 1011,
- "." → 1100,
- "," → 1101,
- "[" → 1110,
- "]" → 1111.

Next, concatenate the resulting binary codes into one binary number in the same order as in the program. Finally, write this number using unary numeral system — this is the Unary program equivalent to the original Brainfuck one.

You are given a Brainfuck program. Your task is to calculate the size of the equivalent Unary program, and print it modulo 1000003 $(10^6 + 3)$.

Input

The input will consist of a single line p which gives a Brainfuck program. String p will contain between 1 and 100 characters, inclusive. Each character of p will be "+", "-", "[", "]", "<", ">", "." or ",".

Output

Output the size of the equivalent Unary program modulo 1000003 $(10^6 + 3)$.

input
```
,.
```
output
```
220
```

input
```
++++[>,.<-]
```
output
```
61425
```

To write a number n in unary numeral system, one simply has to write 1 n times. For example, 5 written in unary system will be 11111.

In the first example replacing Brainfuck commands with binary code will give us 1101 1100. After we concatenate the codes, we'll get 11011100 in binary system, or 220 in decimal. That's exactly the number of tokens in the equivalent Unary program.

T. Cookies

2 seconds, 256 megabytes

Olga came to visit the twins Anna and Maria and saw that they have many cookies. The cookies are distributed into bags. As there are many cookies, Olga decided that it's no big deal if she steals a bag. However, she doesn't want the sisters to quarrel because of nothing when they divide the cookies. That's why Olga wants to steal a bag with cookies so that the number of cookies in the remaining bags was even, that is, so that Anna and Maria could evenly divide it into two (even 0 remaining cookies will do, just as any other even number). How many ways there are to steal exactly one cookie bag so that the total number of cookies in the remaining bags was even?

Input

The first line contains the only integer n $(1 \le n \le 100)$ — the number of cookie bags Anna and Maria have. The second line contains n integers a_i $(1 \le a_i \le 100)$ — the number of cookies in the i-th bag.

Output

Print in the only line the only number — the sought number of ways. If there are no such ways print 0.

input
```
1
1
```
output
```
1
```

input
```
10
1 2 2 3 4 4 4 2 2 2
```
output
```
8
```

input
```
11
2 2 2 2 2 2 2 2 2 2 99
```
output
```
1
```

In the first sample Olga should take the only bag so that the twins ended up with the even number of cookies.

In the second sample Olga can take any of five bags with two cookies or any of three bags with four cookies — $5 + 3 = 8$ ways in total.

In the third sample, no matter which bag with two cookies Olga chooses, the twins are left with $2 * 9 + 99 = 117$ cookies. Thus, Olga has only one option: to take the bag with 99 cookies.

U. Measuring Lengths in Baden

2 seconds, 256 megabytes

Lengths are measures in Baden in inches and feet. To a length from centimeters it is enough to know that an inch equals three centimeters in Baden and one foot contains 12 inches.

You are given a length equal to n centimeters. Your task is to convert it to feet and inches so that the number of feet was maximum. The result should be an integer rounded to the closest value containing an integral number of inches.

Note that when you round up, 1 cm rounds up to 0 inches and 2 cm round up to 1 inch.

Input

The only line contains an integer n $(1 \le n \le 10000)$.

Output

Print two non-negative space-separated integers a and b, where a is the numbers of feet and b is the number of inches.

input
```
42
```

output

```
1 2
```

input

```
5
```

output

```
0 2
```

V. GCD of Polynomials

2 seconds, 256 megabytes

Suppose you have two polynomials and . Then polynomial can be uniquely represented in the following way:

This can be done using long division. Here, denotes the degree of polynomial $P(x)$. is called the remainder of division of polynomial by polynomial , it is also denoted as .

Since there is a way to divide polynomials with remainder, we can define Euclid's algorithm of finding the greatest common divisor of two polynomials. The algorithm takes two polynomials . If the polynomial is zero, the result is , otherwise the result is the value the algorithm returns for pair . On each step the degree of the second argument decreases, so the algorithm works in finite number of steps. But how large that number could be? You are to answer this question.

You are given an integer n. You have to build two polynomials with degrees not greater than n, such that their coefficients are integers not exceeding 1 by their absolute value, the leading coefficients (ones with the greatest power of x) are equal to one, and the described Euclid's algorithm performs exactly n steps finding their greatest common divisor. Moreover, the degree of the first polynomial should be greater than the degree of the second. By a step of the algorithm we mean the transition from pair to pair .

Input

You are given a single integer n $(1 \le n \le 150)$ — the number of steps of the algorithm you need to reach.

Output

Print two polynomials in the following format.

In the first line print a single integer m $(0 \le m \le n)$ — the degree of the polynomial.

In the second line print $m + 1$ integers between -1 and 1 — the coefficients of the polynomial, from constant to leading.

The degree of the first polynomial should be greater than the degree of the second polynomial, the leading coefficients should be equal to 1. Euclid's algorithm should perform exactly n steps when called using these polynomials.

If there is no answer for the given n, print `-1`.

If there are multiple answer, print any of them.

input

```
1
```

output

```
1
0 1
0
1
```

input

```
2
```

output

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

In the second example you can print polynomials $x^2 - 1$ and x. The sequence of transitions is

$$(x^2 - 1, x) \to (x, -1) \to (-1, 0).$$

There are two steps in it.

W. Shovel Sale

1 second, 256 megabytes

There are n shovels in Polycarp's shop. The i-th shovel costs i burles, that is, the first shovel costs 1 burle, the second shovel costs 2 burles, the third shovel costs 3 burles, and so on. Polycarps wants to sell shovels in pairs.

Visitors are more likely to buy a pair of shovels if their total cost ends with several 9s. Because of this, Polycarp wants to choose a pair of shovels to sell in such a way that the sum of their costs ends with maximum possible number of nines. For example, if he chooses shovels with costs 12345 and 37454, their total cost is 49799, it ends with two nines.

You are to compute the number of pairs of shovels such that their total cost ends with maximum possible number of nines. Two pairs are considered different if there is a shovel presented in one pair, but not in the other.

Input

The first line contains a single integer n $(2 \le n \le 10^9)$ — the number of shovels in Polycarp's shop.

Output

Print the number of pairs of shovels such that their total cost ends with maximum possible number of nines.

Note that it is possible that the largest number of 9s at the end is 0, then you should count all such ways.

It is guaranteed that for every $n \le 10^9$ the answer doesn't exceed $2 \cdot 10^9$.

input

```
7
```

output

```
3
```

input

```
14
```

output

```
9
```

input

```
50
```

output

```
1
```

In the first example the maximum possible number of nines at the end is one. Polycarp cah choose the following pairs of shovels for that purpose:

- 2 and 7;
- 3 and 6;
- 4 and 5.

In the second example the maximum number of nines at the end of total cost of two shovels is one. The following pairs of shovels suit Polycarp:

- 1 and 8;
- 2 and 7;
- 3 and 6;
- 4 and 5;
- 5 and 14;
- 6 and 13;
- 7 and 12;

- 8 and 11;
- 9 and 10.

In the third example it is necessary to choose shovels 49 and 50, because the sum of their cost is 99, that means that the total number of nines is equal to two, which is maximum possible for $n = 50$.

X. Dividing the numbers

1 second, 256 megabytes

Petya has n integers: $1, 2, 3, ..., n$. He wants to split these integers in **two non-empty** groups in such a way that the absolute difference of sums of integers in each group is as small as possible.

Help Petya to split the integers. Each of n integers should be exactly in one group.

Input
The first line contains a single integer n ($2 \le n \le 60\,000$) — the number of integers Petya has.

Output
Print the smallest possible absolute difference in the first line.

In the second line print the size of the first group, followed by the integers in that group. You can print these integers in arbitrary order. If there are multiple answers, print any of them.

input
```
4
```
output
```
0
2 1 4
```

input
```
2
```
output
```
1
1 1
```

In the first example you have to put integers 1 and 4 in the first group, and 2 and 3 in the second. This way the sum in each group is 5, and the absolute difference is 0.

In the second example there are only two integers, and since both groups should be non-empty, you have to put one integer in the first group and one in the second. This way the absolute difference of sums of integers in each group is 1.

Y. Splitting in Teams

1 second, 256 megabytes

There were n groups of students which came to write a training contest. A group is either one person who can write the contest with anyone else, or two people who want to write the contest in the same team.

The coach decided to form teams of exactly three people for this training. Determine the maximum number of teams of three people he can form. It is possible that he can't use all groups to form teams. For groups of two, either both students should write the contest, or both should not. If two students from a group of two will write the contest, they should be in the same team.

Input
The first line contains single integer n ($2 \le n \le 2 \cdot 10^5$) — the number of groups.

The second line contains a sequence of integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 2$), where a_i is the number of people in group i.

Output
Print the maximum number of teams of three people the coach can form.

input
```
4
1 1 2 1
```
output
```
1
```

input
```
2
2 2
```
output
```
0
```

input
```
7
2 2 2 1 1 1 1
```
output
```
3
```

input
```
3
1 1 1
```
output
```
1
```

In the first example the coach can form one team. For example, he can take students from the first, second and fourth groups.

In the second example he can't make a single team.

In the third example the coach can form three teams. For example, he can do this in the following way:

- The first group (of two people) and the seventh group (of one person),
- The second group (of two people) and the sixth group (of one person),
- The third group (of two people) and the fourth group (of one person).

Z. Rounding

1 second, 256 megabytes

Vasya has a non-negative integer n. He wants to round it to nearest integer, which ends up with 0. If n already ends up with 0, Vasya considers it already rounded.

For example, if $n = 4722$ answer is 4720. If $n = 5$ Vasya can round it to 0 or to 10. Both ways are correct.

For given n find out to which integer will Vasya round it.

Input
The first line contains single integer n ($0 \le n \le 10^9$) — number that Vasya has.

Output
Print result of rounding n. Pay attention that in some cases answer isn't unique. In that case print any correct answer.

input
```
5
```
output
```
0
```

input
```
113
```
output
```
110
```

input
```
1000000000
```

output

```
1000000000
```

input

```
5432359
```

output

```
5432360
```

In the first example $n = 5$. Nearest integers, that ends up with zero are 0 and 10. Any of these answers is correct, so you can print 0 or 10.