Problem A. Attention! This is Div C/D!

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

If you see this problem as the first problem of the round, then you are in division C/D. If this is not the division you expected to participate in — please log out and enter the correct division immediately!

“`DIVCD`” is not only the designation of the division but also a string of Roman numerals of length 5. The longest Roman numeral in this string (`DIV`=504) has a length of 3. However, a participant in Div C/D must be able to transition from a specific problem to a general one.

The string is not given to us. Only the length of the string n is specified, composed of the letters `I`, `V`, `X`, `L`, `C`, `D`, and `M`. Find the average length of the longest Roman numeral that is a substring of this string.

We remind you of the modern way of writing Roman numerals (the table is taken from Wikipedia):

Place Value	Thousands	Hundreds	Tens	Units
1	M	C	X	I
2	MM	CC	XX	II
3	MMM	CCC	XXX	III
4		CD	XL	IV
5		D	L	V
6		DC	LX	VI
7		DCC	LXX	VII
8		DCCC	LXXX	VIII
9		CM	XC	IX

Note that:

- The numbers 4, 9, 40, 90, 400, and 900 are written in subtractive notation, where the first symbol is subtracted from the second (for example, for 40 (`XL`), ‘`X`’ (10) is subtracted from ‘`L`’ (50)). This is the **only** place where subtractive notation is used.
- A number containing multiple decimal digits is constructed by appending the Roman equivalent of each digit from the most significant to the least significant.
- If the decimal digit is 0, no digits are written in the Roman representation for that place.
- The largest number that can be represented in the Roman numeral system is 3,999 (MMMCMXCIX).

Input

The input contains a single integer n ($1 \le n \le 1000$) — the length of the string.

Output

It can be shown that the answer can be uniquely represented as a rational number p/q, where p and q are coprime positive integers. Output the number $p \cdot q^{-1} \bmod 998\,244\,353$, that is, the remainder of the division of p by such a number q', that $q \cdot q' - 1$ is divisible by $998\,244\,353$.

Examples

standard input	standard output
2	509308345
1	1

Problem B. BA Stacks

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Two players are playing a game. Initially, they have three non-empty stacks containing letters, and the letters can only be "A" and "B". It is also known that no stack contains two adjacent identical letters. Players take turns making moves. On their turn, a player must take the top letter from any non-empty stack and place it on top of another stack such that the top of that stack had the same letter before the move. After that, these two equal letters are removed from the top of the stack where the letter was placed. Your task is to determine who will win with optimal play from both players—the player making the first move or the player making the second move.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows. Each test case consists of three lines, each containing information about the corresponding stack. In the i-th ($1 \le i \le 3$) line, there are two numbers top_i and n_i ($top_i =$ "A" or $top_i =$ "B", $1 \le n_i \le 10^9$), representing the symbol on the top of the i-th stack and the number of elements in it. Since the symbols alternate, this information uniquely defines the stack.

Output

For each test case, output 1 if the first player wins for that case, and 0 otherwise.

Example

standard input	standard output
5 A 1 A 1 A 1 A 1 A 2 A 3 A 5 B 90 A 11 B 10 B 10 B 100 A 155 B 155 A 15	1 1 1 0 0

Problem C. Continuous Sums of Digits

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 1024 megabytes

Let $S(x)$ denote the sum of the digits of the positive integer x. Define $S^0(x) = x$, $S^1(x) = S(x)$, $\ldots$, $S^k(x) = S(S^{k-1}(x))$. You are given the numbers n and k. Find all solutions to the equation

$x + S(x) + S^2(x) + S^3(x) + \ldots + S^k(x) = n.$

Input

In the first line, you are given the number of test cases $1 \le T \le 500$. In the following T lines, each test is described by a pair $1 \le n \le 10^{12}$, $0 \le k \le 10^{10}$.

Output

For each test, output all solutions to the equation on a separate line, separated by spaces in ascending order. Start the output with the number of solutions, followed by all solutions. The solutions output for each test must be distinct.

Example

standard input	standard output

3
33 2
1654 4
992 3
	2
	17 23
	1
	1625
	1
	959

Problem D. Datacenter And Logical Operations

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

In modern cloud infrastructure and datacenter control systems, particularly in MWS CDN, performance at the basic query level is of great importance.

You have been invited for an internship at MTS and tasked with solving the following problem related to low-level optimization of system architecture.

You have a set of n zeros and m ones, as well as a set of a logical operations OR and a set of $n+m-1-a$ logical operations AND. First, you write down all the zeros and ones you have in a row, and then above each pair of adjacent elements in the row, you write one of the logical operations you have. You must use exactly as many operations of each type as you have available. After that, you calculate the result of the resulting expression as follows: for each pair of adjacent elements in the row, the operation written above it is applied, and the result of this operation is added to the total result. The results of the operations are considered as the results of classical logical operations "AND" and "OR", namely:

- $0\ OR\ 0 = 0$
- $0\ OR\ 1 = 1$
- $1\ OR\ 0 = 1$
- $1\ OR\ 1 = 1$

- $0\ AND\ 0 = 0$
- $0\ AND\ 1 = 0$
- $1\ AND\ 0 = 0$
- $1\ AND\ 1 = 1$

Your task is to maximize the final result by optimally arranging the numbers and operations.

Input

The first line contains the number $1 \le T \le 10^4$ – the number of test cases. The following T lines contain independent queries – each line gives you three numbers $0 \le n \le 10^9$, $0 \le m \le 10^9$, and $0 \le a \le n+m-1$. It is guaranteed that $n + m > 0$.

Output

For each query, output the answer on a separate line – the maximum result if the numbers and operations are arranged optimally.

Example

standard input	standard output

2
1 8 7
100 0 0
	8
	0

Problem E. Easier Division

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

The organizers of the RuCode festival have to check long lists on the day of the competition, verifying whether a participant has passed the selection and in which division they are participating.

You decided to develop a system that would immediately allow recognizing div EF. Since participants will only be able to practice in the next season, you decided to practice with numbers. Your task is to check a number for compliance with div EF. A number corresponds to div EF if it is divisible by a number that corresponds to the representation EF in some numeral system (it is assumed that numeral systems use the digits 0-9 first, then uppercase letters from `A` to `Z`; for all other digits less than the base, some symbols also exist, but they are not important for this task).

For example, 42 is not a div EF number, while 478 is (since it is divisible by $239 = EF_{16}$).

In case the number is a div EF number, find the minimum base of the numeral system in which one of the divisors of this number has the form EF.

Input

The first line of input contains one integer n ($1 \le n \le 10^{12}$).

Output

If the given number n is a div EF number, output the minimum base of the numeral system in which one of the divisors of this number has the form EF. Otherwise, output -1.

Examples

standard input	standard output
42	-1
478	16

Problem F. Fine Triangles

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

We will call *fine* right triangles that have the following property: all sides of the triangle have integer lengths, and the perimeter of the triangle is divisible by at least one of its legs. For example, the classic Egyptian triangle has a perimeter of $3+4+5=12$ and is fine: its perimeter is divisible by 3.

Your task is to determine how many different fine triangles exist where one of the legs has a length of n.

Input

The input consists of a single integer n ($1 \le n \le 10^{12}$).

Output

Output a single integer — the number of different fine triangles where one of the legs has a length of n.

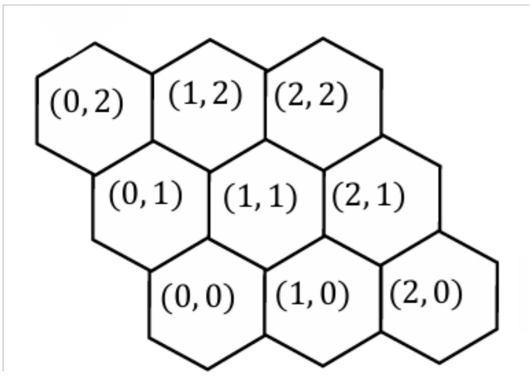
Example

`standard input`	`standard output`
`6`	`1`

Problem G. Game On Hexagonal Board

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

In the computer game “Mighty Mages and Heroes”, the game field consists of $(w+1) \times (h+1)$ hexagonal cells. You play as a dragon, your dragon is located at cell $(0,0)$, and its cave is at cell (w,h). The image below shows an example of the numbering of the bottom left corner of the field (the neighborhoods of cell $(0,0)$).



If the dragon is in cell (x,y), it can move to one of the cells $(x-1,y-1)$, $(x-1,y)$, $(x,y-1)$, $(x+1,y)$, $(x,y+1)$, $(x+1,y+1)$, while cells with $x<0$, $x>w$, $y<0$, and $y>h$ do not exist (and the dragon cannot move to such a cell).

At the same time, containers with gasoline are placed on the cells. Initially, the fire hazard level of all cells is 0. If a container with a volume of a liters is placed in cell (x,y), the fire hazard level changes as follows.

Let j be the minimum number of boundary crossings when moving from cell (x,y) to cell (u,v). Then, if $a-j$ is positive, the fire hazard level of cell (u,v) increases by $a-j$. If $a-j$ is less than or equal to 0, the fire hazard level of cell (u,v) does not change.

A total of n containers with gasoline were placed one after another, the i-th container is placed in cell (x_i,y_i) and will have a volume of a_i liters.

You want to guide the dragon along a route such that the maximum fire hazard level f_{max} among all cells it passes through is minimized.

Input

The first line of input contains two integers w and h — the dimensions of the field ($1 \le h,w \le 2\cdot 10^5$ and $h \times w \le 2\cdot 10^5$) and the number of containers with gasoline n ($1 \le n \le 2\cdot 10^5$).

The i-th of the following n lines specifies the coordinates of the i-th container x_i ($0 \le x_i \le w$), y_i ($0 \le y_i \le h$) and its volume a_i ($1 \le a_i \le 10^9$) for the i-th placement.

Output

Output a single number — the maximum fire hazard level f_{max}.

Example

standard input	standard output

6 5 5
1 2 3
6 0 5
4 1 1
4 4 4
6 5 1
	3

Problem H. Hidden Swamp

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

In a hidden swamp, n lily pads are arranged in a circle. We number them clockwise from 1 to n. Two frogs — Anfisa and Vasilisa — are sitting on two of these lily pads, with the first frog sitting on lily pad a and the second frog sitting on lily pad b. They need to meet, meaning they must end up on the same lily pad. To do this, they can make the following moves (a move consists of moving to an adjacent lily pad):

- Anfisa moves clockwise and takes t_1 seconds
- Anfisa moves counterclockwise and takes t_2 seconds
- Vasilisa moves clockwise and takes t_3 seconds
- Vasilisa moves counterclockwise and takes t_4 seconds
- Both frogs move towards each other — Anfisa moves clockwise, and then Vasilisa moves counterclockwise. This action is not allowed if after Anfisa's move she ends up on the same lily pad as Vasilisa before the move — since the second must move. This action takes a total of t_5 seconds
- Both frogs move towards each other — Anfisa moves counterclockwise, and then Vasilisa moves clockwise. This action is not allowed if after Anfisa's move she ends up on the same lily pad as Vasilisa before the move — since the second must move. This action takes a total of t_6 seconds

Your task is to determine the minimum time in seconds after which the two frogs can meet if they act optimally.

Input

In the first line, you are given the number of queries in one test $1 \le T \le 10^4$. In the following T lines, the queries themselves are given. Each query is characterized by nine numbers — the number of lily pads $1 \le n \le 10^9$, the positions of the frogs $1 \le a, b \le n$ (first the position of Anfisa, then Vasilisa), and the times $1 \le t_1, t_2, t_3, t_4, t_5, t_6 \le 10^9$. Each query represents an independent problem that you must solve.

Output

For each query, output the minimum time in seconds required for the frogs to meet on a separate line.

Example

standard input	standard output

`3`
`8 7 2 4 1 2 3 3 2`
`100 15 61 8 11 14 7 6 6`
`448 1 448 1 10000 10000 1 2 1000000`
	`5`
	`138`
	`447`

Problem I. Interactive GCD game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

This is an interactive problem

Ekaterina is in the seventh grade and is an enthusiast of competitive programming. So she not only actively practices solving problems to become the absolute winner of the IOI, but she also knows the history of the Olympiads well, in particular, the fact that the first Soviet Olympiad in Informatics took place in the 1987-1988 academic year. Therefore, Ekaterina considers all years from 1 to 1986 inclusive as *prehistoric*.

Before a team training session, Ekaterina suggested playing the following game: she thinks of a "prehistoric"year (that is, an integer from 1 to 1986 inclusive), and the rest of the team makes queries.

Each query consists of an integer k, also within the range from 1 to 1986 inclusive. In response, Ekaterina calculates the greatest common divisor d of the numbers n and k, checks if d is written on a piece of paper (the paper is empty at the beginning of the game), if it is written — she says "yes otherwise, she writes d on the paper and responds "no".

To win, the team must guess the number in 300 or fewer attempts.

The teammates are less experienced in competitive programming than Ekaterina and ask you to help solve the problem.

Interaction Protocol

The interaction begins with the jury program, which plays for Ekaterina. It outputs a single integer t — the number of scenarios ($1 \le t \le 50$).

Then, in each scenario, your program starts the interaction. To make a query, output a question mark (`?`), followed by a space, and then the number k ($1 \le k \le 1986$). The jury program will output 1 if the greatest common divisor n and k has already been encountered previously as a response to some of the earlier queries, and 0 otherwise.

To respond to a query, output an exclamation mark (`!`), followed by a space, and then the number n. When counting the number of queries, the response output is not counted as a query.

If you stay within the required number of queries and the number n is guessed correctly, the jury program moves on to the next scenario. If something goes wrong, you will receive a verdict from the testing system about the error.

It is guaranteed that the jury program does not change the number n after the interaction process begins (that is, the interactor is not adaptive).

Example

standard input	standard output
1	
	? 1
0	
	? 995
1	
	? 993
1	
	? 991
0	
	? 2
0	
	! 1982

Note

In the example from the statement, the first response in the list is one. Then the greatest common divisor for the numbers 995 and 993 is already in the list (that is, it equals 1), the greatest common divisor for 991 is not in the list, meaning it is not equal to 1; therefore, it equals 991 (since 991 is prime). There are only 2 numbers that are multiples of 991, so we check divisibility by 2 and find the answer 1982.

Problem J. Judges vs Cheaters

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

This is a double-run problem

...During the broadcast of a competitive programming event, the studio guest — a well-known coach in niche circles — received an anonymous message: “If you want to know the difficulty of the problems in the upcoming round, go to room 322 during the commercial break”. This was about the final round of the competition, which was planned to have 10 problems. It could also have been a provocation – any attempt to contact the team would lead to disqualification. The coach looked at the interview program — after the break, the host planned to ask questions about two recent rounds of ForceCoders — one round had 7 problems, and the other had 5, and also to ask for a prediction of which of the 10 problems in the upcoming final would be solved first. The coach got an idea how to pass the information and contacted the team before going to room 322.

...In room 322, the coach indeed saw a list of 10 problems sorted by difficulty. The problems, as usual, were numbered with uppercase English letters from `A` to `J`. The coach memorized the list and returned to the studio. The host then asked the last three questions — requested to order the problems from the rounds of seven and five problems by originality, and also asked which problem from the 10 would be solved first in the upcoming final. The coach could choose the order in which to answer these questions. The answers to the first and second should be the ordered letters of the problems, and the answer to the third should be one letter from `A` to `J`.

However, vigilant organizers, suspecting the purpose of the visit to room 322, "bleeped"the letter of the problem named by the coach. Nevertheless, the team watching the broadcast was able to reconstruct the required order of the problems from this fragment of the interview.

...You are tasked, as part of an internal investigation by the jury into the information leak, to repeat what the coach and the team did and convey the order of the problems using three messages — transmitted in the order of your choice: a string of length 7 with the names of the problems from `A` to `G`, arranged in some order, a similar string with the names of the problems from `A` to `E`, and one letter that will be "bleeped"during the broadcast.

Input

Your program will be run twice, first in `coach` mode, and then in `team` mode.

If the problem is run in `coach` mode, the first line of input will be `coach`. This is followed by one integer t ($1 \le t \le 10^5$) — the number of test cases. Each test case consists of one string of length 10, made up of uppercase Latin letters from `A` to `J`, where each letter appears exactly once.

If the problem is run in `team` mode, the first line of input will be `team`. This is followed by one integer t ($1 \le t \le 10^5$) — the number of test cases. Each test case is on one line and consists of three words separated by spaces — the coach’s answers to the questions in the broadcast. One line contains 7 uppercase Latin letters from `A` to `G`, where each letter appears exactly once, the second contains 5 uppercase Latin letters from `A` to `E`, where each letter appears exactly once, and the third contains the letter `A` (regardless of which letter was named by the coach). The order of the blocks is preserved.

Output

If the problem is run in `coach` mode, for each test case you should output three lines. One line of 7 uppercase Latin letters from `A` to `G`, where each letter appears exactly once, the second line of 5 uppercase Latin letters from `A` to `E`, where each letter appears exactly once, and the third line containing one letter from `A` to `J`. The order of the blocks is determined by your program.

If the problem is run in `team` mode, output one line for each test case — the reconstructed order of the problem difficulties in the upcoming round, that is, a string of length 10 that was obtained in `coach` mode.

Examples

standard input	standard output
coach 1 ABCDEFGHIJ	J BEDACFG CEBAD
team 1 A BEDACFG CEBAD	ABCDEFGHIJ

Problem K. Kingminer

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

For the presentation of a large project on the development of the rare minerals, top manager of the Kingminer company Bogdan received a sequence of non-negative integers $a_1, a_2, \ldots, a_n$. Some elements in the sequence are equal to zero — this means that the required parameter has not yet been determined. After consulting with LLM MWS Copilot, Bogdan decided that it is worth replacing the zeros with some integer x.

An experienced investor immediately notices the potential for growth. Therefore, the impression that the presentation makes on investors depends on the length of the longest strictly increasing subsequence. We denote the dependence of this length on the value of x as $f(x)$.

It is required to find $\sum_{i=1}^{n} i \cdot f(i)$.

Recall that the length of the longest increasing subsequence of the sequence $s_1, s_2, \ldots, s_m$ is the largest k such that there exist $1 \le i_1 < i_2 < \cdots < i_k \le m$ satisfying $s_{i_1} < s_{i_2} < \cdots < s_{i_k}$.

Input

The input consists of several test cases.

Each test case consists of two lines. The first line contains a single integer n — the length of the sequence a ($1 \le n \le 10^5$). The second line contains n integers $a_1, a_2, \ldots, a_n$ — the elements of the sequence ($0 \le a_i \le n$).

It is guaranteed that the sum of n across all test cases does not exceed $2.5 \cdot 10^5$.

The input ends with a test case where $n = 0$, which should not be processed.

Output

For each test case, output a single integer on a separate line — the answer to the problem.

Example

standard input	standard output
2	3
1 1	24
4	56
2 0 3 0	3
6	
3 0 6 1 0 4	
2	
0 0	
0	

Problem L. Long Array and XOR

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Alex has an array of n non-negative numbers $a_1, a_2, ..., a_n$. Let $f(x)$ denote the smallest number of the form 2^k, where $k \geq 0$, that is strictly greater than x.

Alex can perform certain operations on the array. In one operation, he can choose some $1 \leq i \leq n$ and replace a_i with $f(a_i) - 1 - a_i$. There is an additional constraint: for each index i, he can apply the operation to the number at that index no more than once.

Alex wants the bitwise exclusive OR of the resulting elements of the array to be as large as possible. Help him understand what the maximum value he can achieve is.

Answer q queries. In the i-th query, you are given two numbers l_i and r_i, and you independently solve the problem for the subarray of the original array, where the elements are taken from index l_i to index r_i.

Input

In the first line, you are given a number $1 \leq n \leq 2 \cdot 10^5$, which denotes the number of elements in Alex's array, and a number $1 \leq q \leq 2 \cdot 10^5$, which denotes the number of queries. In the next line, you are given the elements of the array – non-negative integers $0 \leq a_i \leq 10^9$. In the following q lines, each query is described by two numbers $1 \leq l_i \leq r_i \leq n$.

Output

For each query, output the answer on a separate line.

Example

standard input	standard output

8 5
3 5 7 6 2 4 3 2
1 8
2 4
2 8
3 7
4 8
	7
	4
	7
	7
	6

Problem M. Minute

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

After the tour of the RuCode championship, participants from all divisions lined up for a photo on one of the venues. As is known, each participant belongs to one of six divisions from `A` to `F`.

According to the local organizers' plan, each participant holds a sign with the letter of their division. "Wait a minute!" said the photographer...

Observing the process from the side, the coach of one of the teams wondered: if we consider the letters on the participants from left to right as a hexadecimal number representing time in seconds, what is the minimum number of participants that should approach so that the time becomes a whole number of minutes, and from which divisions these participants will be? Participants can approach from either the right or the left, but cannot stand between other participants.

Input

The first line of input consists of letters from `A` to `F`. The string is non-empty and its length does not exceed 10^5.

Output

If the string, interpreted as a hexadecimal number, is already divisible by 60, output `OK`. Otherwise, output a string composed of the minimum number of division letters that lists the participants who can be added to the arrangement from its ends so that the number becomes divisible by 60. If there are multiple such strings, output the lexicographically smallest one. The order of the letters in the string does not have to match the order in which participants will be added to the arrangement.

Examples

standard input	standard output
`ABCD`	`AACC`
`ABEDC`	`OK`