Problem A. An Experiment in Optics Lab

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

Maria is a scientist in a laboratory. At the moment, she conducts experiments with lasers and the refraction of different materials. She assembled n materials in a single line. Each material has length ℓ_i and refractive index r_i. Now Maria wants to perform the following operations q times:

- She substitutes the i-th material with another one.
- She shines the laser from some point of the line. The initial angle between the ground and the laser beam is given.

For the purpose of our problem, we can model all materials as vertical strips. The laser starts at $y = 0$ and some given x.

Maria is interested in the y-coordinate of the point at which the laser beam will escape from the last material.

Recall that if a laser beam passes the boundary between two materials with refractive indices r_1 and r_2, then the relation between the angles θ_1 and θ_2 between the laser beam and the horizontal line is $r_1 \sin\theta_1 = r_2 \sin\theta_2$.

Input

The first line contains one integer n ($1 \le n \le 10^5$). The next n lines contain the description of initial materials. Each material is described by two integers — its length ℓ_i and its refractive index r_i ($1 \le \ell_i \le 1000$, $10^4 \le r_i \le 1.6 \cdot 10^4$).

The next line contains one integer q ($1 \le q \le 10^5$). The following q lines contain the description of the experiment. Each line starts with an integer t ($1 \le t \le 2$).

If $t = 1$, then it is followed by three more integers, id, ℓ, r ($1 \le id \le n$, $1 \le \ell \le 1000$, $10^4 \le r \le 1.6 \cdot 10^4$). This means that Maria wants to replace the id-th material with another material with length ℓ and refractive index r.

If $t = 2$, then it is followed by two integers, x and ang. This means that Maria wants to shine the laser from point $(x, 0)$ with angle ang and wants to know the y-coordinate of the laser beam when it escapes the last material in the line. It is guaranteed that x is non-negative and is strictly less than the sum of all lengths of materials. If the starting point lies on the border of two materials, we assume that the laser starts in the material with a bigger index. The angle ang is given in seconds, that is, to convert it to degrees one should divide it by 3 600. It is guaranteed that $0 \le ang \le 1.08 \cdot 10^5$.

Output

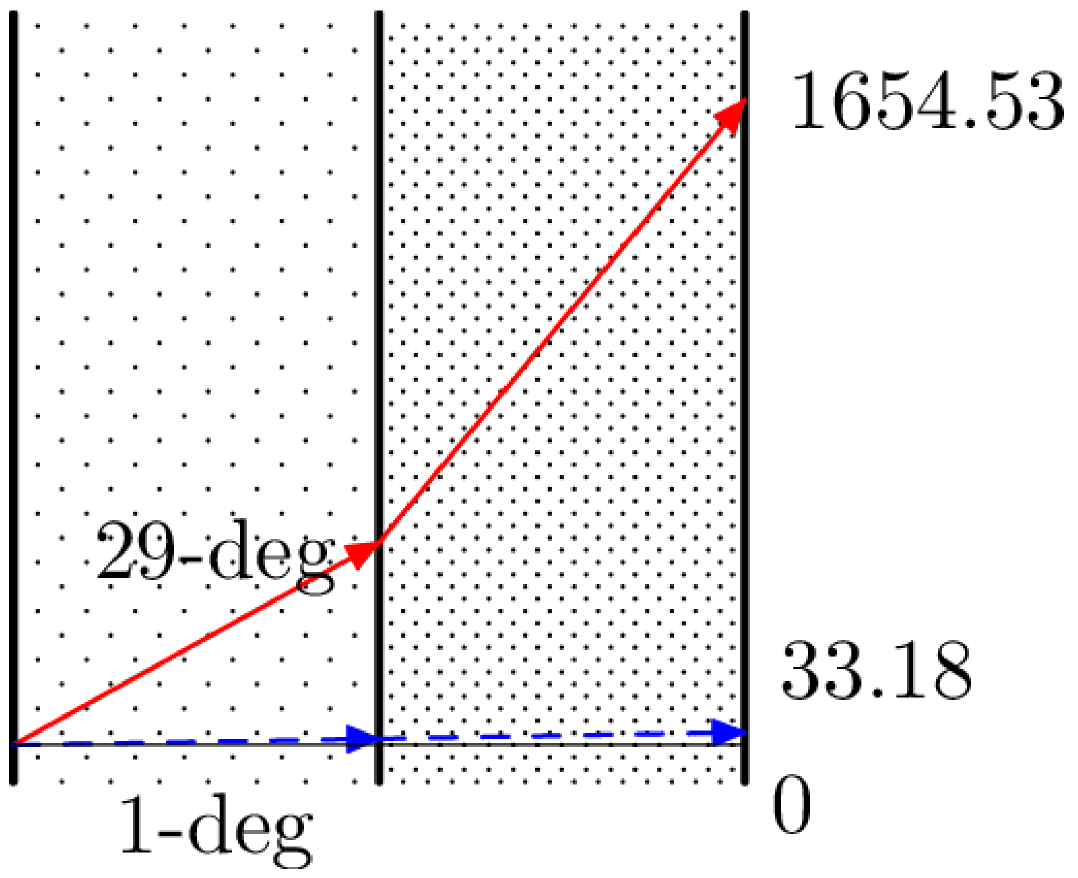
For each event with $t = 2$, print one real number — the answer to the problem. The answer will be considered correct if and only if the absolute or relative error does not exceed 10^{-6}.

It can be proved that under the constraints of the problem the beam will never reflect from the boundary between two materials (i.e. $|\sin\theta|$ will never exceed 1).

Examples

standard input	standard output
2 314 13579 927 13579 3 2 73 79200 1 2 406 13579 2 73 79200	471.90263177546315567779 261.40496811534645696384
2 983 15900 917 10000 3 2 0 104400 1 2 917 15887 2 0 3600	1654.53227745170376805639 33.17772497531446945426

Note



The picture above illustrates the second example. The red (thick solid) line shows the trajectory of the laser beam from the first event, the blue (thick dashed) line shows the trajectory of the laser beam from the third event.

Problem B. Banners Before QC

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

Two stage coordinators, Mira and Leo, are preparing the opening ceremony of a city festival. They still have some jobs to finish, and the guests will arrive very soon! Fortunately, the plan is already fixed, the jobs are split between Mira and Leo, and the time needed for each job is known. The only thing left is to decide in which order they will do their own jobs. Neither coordinator likes handling several jobs at the same time, so each of them will choose an order for their jobs and then complete them one by one in that order.

There is one important rule. Some jobs are "decoration jobs", and some jobs are "technical-check jobs". A decoration job prepares some part of the venue, while a technical-check job verifies that the already prepared parts work as intended. For safety reasons, every decoration job, even one assigned to the other coordinator, has to be completed strictly before any technical-check job is completed. Note that a coordinator may start a technical-check job before all decoration jobs are completed: they can check the parts that are already ready, but the check cannot be fully completed before the whole venue is decorated.

You are the chief coordinator of the ceremony, and your goal is to make sure that Mira and Leo do not waste any time. This means that immediately after one of them completes some job, they have to start doing some other job assigned to them (unless they have already completed all their jobs, of course). Moreover, both coordinators have to start working immediately and simultaneously. Each coordinator is free to choose any order of the jobs assigned to them, but the rule about decoration and technical-check jobs must be satisfied.

You wonder how many correct ways for the coordinators to order their own jobs there are in total. As this number might be very large, calculate it modulo 998 244 353. Two ways are considered different if and only if there are two jobs assigned to one coordinator such that in the first order the first job is completed before the second job and in the second order the second job is completed before the first one.

Input

The first line contains two integers n and m ($1 \le n, m \le 10^5$) — the number of jobs assigned to Mira and Leo, respectively.

The next n lines contain the description of Mira's jobs. Each line contains an integer t_i ($1 \le t_i \le 10^9$), and a letter c_i ($c_i \in \{\texttt{D}, \texttt{T}\}$). The integer is the time to complete this job, and the letter is "`D`" if this job is a decoration job and "`T`" if this job is a technical-check job.

The final m lines contain the description of Leo's jobs in the same format.

Output

Print one integer — the number of correct ways taken modulo 998 244 353.

Examples

standard input	*standard output*
2 2 8 T 100 T 3 D 5 D	2
2 1 10 D 3 T 1 D	1

Problem C. Choosing Best Friend

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

There are n people. Each two people among them are either friends or not. Friendship is bidirectional.

Each person wants to select best friends — some non-empty subset of their friends. The only condition they want to satisfy is that all sets of best friends among n people must be unique. Note that the "being the best friends" property may not be bidirectional (i.e., it is possible that X is best friend for Y, but Y is not the best friend for X).

You are given all friendships. Find any possible selection of sets of best friends such that the total size of sets of best friends is minimum possible. Or you should state that it is impossible to find such a selection.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of testcases. Next lines contain descriptions of testcases.

The first line of each testcase contains a single integer n ($2 \le n \le 500$) — the number of people.

Then $n-1$ lines follow, the i-th line contains a string of length $n-i$ consisting of characters 0 and 1. For each j ($i < j \le n$), the $(j-i)$-th character of this string is 1 if i and j are friends, and 0 otherwise.

It is guaranteed that the sum of n^2 does not exceed $2.5 \cdot 10^5$.

Output

Print answers to testcases in the given order.

If it is impossible to select sets of best friends to satisfy the conditions, print a single integer -1.

Otherwise, print n lines. The i-th line should start with s_i ($s_i \ge 1$) — the number of selected best friends for the i-th person. Then s_i different integers $a_{i,1}, \ldots, a_{i,s_i}$ ($1 \le a_{i,j} \le n$, $a_{i,j} \ne i$) in the line should follow — best friends of the i-th person. For each j ($1 \le j \le s_i$) people i and $a_{i,j}$ should be friends.

All sets $\{a_{i,1}, \ldots, a_{i,s_i}\}$ should be different. The sum $\sum_{i=1}^{n} s_i$ should be minimum possible.

If there are multiple possible answers, you should print any.

Example

standard input	*standard output*
2 5 1000 011 10 1 3 11 0	1 2 1 1 1 4 1 3 2 2 4 -1

Problem D. Deletion-Proof Badge

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

The security office of the city festival prints a positive integer on every access badge. Before a badge is approved, an inspector may erase zero or more digits from its decimal representation, keep the remaining digits in their original order, and read the result as a decimal integer. At least one digit must remain. If the inspector can obtain a prime number this way, the badge code is considered suspicious.

Call a positive integer *prime-proof* if it is not prime itself and no prime number can be obtained from its decimal representation by erasing digits. For example, 104 is prime-proof because neither 0, 1, 4, 10, 14, nor 104 are primes, but 2024 is not prime-proof because we can erase the first, second, and fourth digits and obtain the prime number 2.

Note that the prime numbers are the integers that have exactly two distinct divisors: themselves and one.

Given an integer n, count the number of prime-proof codes with exactly n decimal digits (without leading zeroes) modulo 998 244 353.

Input

Input contains one integer n ($1 \le n \le 10^6$).

Output

Print one integer — the number of n-digit prime-proof codes modulo 998 244 353.

Example

standard input	standard output
1	5

Note

The one-digit prime-proof codes, mentioned in the sample, are 1,4,6,8, and 9.

Problem E. Expose The Werewolf

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	1024 mebibytes

You are playing the Werewolf game with your friends. In this game, there are a total of n participants numbered from 1 to n, your number is 1. All participants are divided into two teams — players of one team are werewolves, and players of another team are villagers. The villagers do not know who the werewolves are, and their goal is to determine it.

You are a villager and you have a special role: the journalist. You can ask about any two players (except yourself) and you will know whether they are both from the same team (werewolves or villagers), but you won't know their team.

You know that there are a total of m werewolves, and you want to determine them. During the game, you have already asked k questions. In the i-th question, you have asked about players a_i and b_i, and you know whether they are in the same team. You can ask at most 2 questions before the game finishes for you; can you determine who is in the werewolves team?

Input

The first line of the input contains an integer t ($1 \le t \le 1000$) — the number of testcases.

The first line of each testcase contains three integers n, m, and k ($1 \le m < n \le 5 \cdot 10^5$, $0 \le k \le 5 \cdot 10^5$) — the number of players, the number of players in the werewolves team, and the number of questions you have asked.

Each of the next k lines of the testcase contains three integers a_i, b_i and c_i ($2 \le a_i, b_i \le n$, $0 \le c_i \le 1$). If players a and b are from the same team, then $c_i = 1$; otherwise, $c_i = 0$.

You may assume that:

- all answers are correct and there exists a werewolves team of m members;
- the sum of n in all testcases does not exceed $5 \cdot 10^5$;
- the sum of k in all testcases does not exceed $5 \cdot 10^5$.

Output

For each testcase, print "`Yes`" if it is possible to determine all players in the werewolves after asking no more than two queries; otherwise, print "`No`".

Example

standard input	*standard output*
5 7 2 5 2 3 1 2 4 0 3 4 0 4 5 1 6 7 1 3 1 0 9 2 3 2 3 1 5 6 1 8 9 1 5 1 0 8 3 3 2 3 0 4 5 0 5 4 0	Yes No Yes Yes No

Problem F. Fix The Bad Ping

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

This is an interactive problem.

Alber Blanc is a proprietary trading firm that provides liquidity for many exchanges with a particular focus on emerging markets. They also provide professional services as a derivatives market maker on different exchanges.

As a developer at Alber Blanc, you have been tasked with optimizing the connections between exchange servers and fund servers. There are a total of 10^9 exchange servers and 10^9 fund servers, and each exchange server is directly connected to every fund server, resulting in 10^{18} connections. Testers have noted that two connections are performing slower than expected. Typically, the latency between servers is 1ms; however, in two cases, the latency is 2ms. To resolve the issue, testers need your help to identify which connections are slow.

To find the slow connections, you can measure the total latency of certain connections. Specifically, you can select e_ℓ, e_r, f_ℓ, and f_r, run tests with these parameters, and determine the total latency of all connections between exchange servers numbered $e_\ell \le i \le e_r$ and fund servers numbered $f_\ell \le j \le f_r$.

The testers urgently request information about the slow connections, so you may measure the total latency no more than 125 times.

Interaction Protocol

The interaction begins with your program making queries. Each query is the line in the format "`?` e_ℓ f_ℓ e_r f_r" (without quotes), where $1 \le e_\ell \le e_r \le 10^9$, $1 \le f_\ell \le f_r \le 10^9$, to represent the parameters of the test. In response, the jury program outputs a single integer — the total latency of selected connections. You may make no more than 125 queries in total.

To output your answer, print the line "`!` e_1 f_1 e_2 f_2" (without quotes), where e_1 and f_1 are the server numbers that make up the first slow connection, and e_2, f_2 are the server numbers of the second slow connection. Connections can be output in any order. Note that outputting the answer does not count as a query.

It is guaranteed that slow connections do not change during the interaction (i.e., the interactor is not adaptive). Remember to print the newline and flush the output buffer after each query and after the answer, or your solution will get `IL` (`Idleness Limit Exceeded`).

Example

standard input	*standard output*
	`? 1 1 1 1`
`1`	
	`? 2 2 2 2`
`2`	
	`? 3 4 5 6`
`9`	
	`? 4 4 6 6`
`10`	
	`? 6 4 6 6`
`4`	
	`? 6 5 6 6`
`2`	
	`! 2 2 6 4`

Problem G. Good Coach

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

On one of the planets in a far, far away galaxy, a numeral system with base b is used. In one of the major cities on this planet, there is a developed network of buses. The routes of the buses are numbered with positive integers.

One day, the Intergalaxy Collegiate Programming Contest (ICPC) Finals was held in this city. The coach of the team from Earth noticed an interesting fact: at the stop near the contest venue, there are x bus route numbers written on the banner, and each of the b digits appears on this banner exactly once.

The coach found that this fact is funny and asked his team to check for given x and b whether such a situation is possible, and if it is, output the b-ary representation of the minimum possible value of the largest number to be written on the banner in this case.

Input

The first line contains one integer x — the number of bus routes whose numbers are displayed on the banner ($1 \le x \le 100$). The second line contains one integer b ($2 \le b \le 100$) — the base of the numeral system.

Output

If the situation is impossible, output -1. Otherwise, output the b-ary representation of the minimum possible value of the largest bus route number displayed on the banner, as a sequence of integers representing the corresponding digits in decimal notation, ordered from the most significant b-ary digit to the least significant. For example, the number $\mathtt{CD3}_{16}$ should be output as "`12 13 3`".

Examples

standard input	*standard output*
10 10	-1
9 10	1 0

Note

In the first example, it is impossible to form 10 positive numbers from the 10 digits of the decimal system in the required way, as 0 is not a positive number. In the second example, the list of routes will be 10, 2, 3, 4, 5, 6, 7, 8, 9.

Problem H. Hall of Prizes

Input file: *standard input*
Output file: *standard output*
Time limit: 3 seconds
Memory limit: 1024 mebibytes

Hannah is visiting the city festival's Hall of Prizes. The hall has several attractions; their exact rules are not important to us. It is enough to know that the attractions are numbered from 1 to n and can be completed in this order.

Hannah has a balance of festival tokens, which initially contains 0 tokens. After completing the i-th attraction, a_i tokens are added to her balance.

Additionally, after completing each attraction, Hannah gains access to a prize booth. The contents and the meaning of this booth are not important to us; it is enough to know that its offer can be refreshed.

For each visit to the prize booth, the first refresh costs c_1 tokens, the second c_2 tokens, and so on. Moreover, $1 \le c_1 \le c_2 \le \ldots \le c_m$. During any visit to the booth, Hannah can make any number of refreshes from 0 to m, but her balance cannot become negative at any moment in time.

It is also possible to skip attractions. If Hannah skips the i-th attraction, she does not receive a_i tokens for it and does not gain access to the prize booth.

Hannah is very interested in the maximum number of refreshes she can make in total across all booth visits if she skips the first t attractions, for each t from 0 to $n-1$. Unfortunately, she is not a programmer, so she turned to you for help.

Input

In the first line of input data, there are two integers n, m ($1 \le n, m \le 3 \cdot 10^5$) — the number of attractions in the hall and the maximum number of refreshes that can be made during one booth visit.

In the second line, there are n integers $a_1, \ldots, a_n$ ($1 \le a_i \le 10^{12}$) — the number of tokens received for completing each attraction.

In the third line, there are m integers $c_1, \ldots, c_m$ ($1 \le c_i \le 10^{12}$, $c_1 \le c_2 \le \ldots \le c_m$) — where c_i is the price of the i-th refresh during each booth visit.

Output

Output n integers — the maximum number of refreshes across all booth visits that can be made if the first t attractions are skipped, for each t from 0 to $n-1$.

Example

standard input	*standard output*
5 4 6 2 12 6 3 1 3 3 5	13 10 9 4 1

Note

In the first example, if Hannah does not skip any attractions, she will be able to make a total of 13 refreshes across all booth visits, following this scheme:

- After the first attraction, make 2 refreshes, spending $1 + 3 = 4$ tokens and leaving 2 tokens in her balance.
- After the second attraction, make another 2 refreshes, spending $1 + 3 = 4$ tokens and leaving 0 tokens in her balance.
- After the third attraction, make another 3 refreshes, spending $1 + 3 + 3 = 7$ tokens and leaving 5 tokens in her balance.

- After the fourth attraction, make another 3 refreshes, spending $1 + 3 + 3 = 7$ tokens and leaving 4 tokens in her balance.
- After the fifth attraction, make another 3 refreshes, spending $1 + 3 + 3 = 7$ tokens and leaving 0 tokens in her balance.

Problem I. Inconvenient Journey: IJ

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

The Berland transport company (BTC) operates express train lines between n cities. In total, there are m bidirectional train lines in the BTC timetable. Each line connects two cities, and it is guaranteed that it is possible to travel between any two cities using the BTC lines with zero or more transfers.

The railway station of each city consists of m platforms. The platforms are numbered with integers from 1 to m. The departure and arrival platforms for each train line are predefined, meaning that a line is specified by two pairs "city and the platform id of that city's station". Note that some platforms may not be present in the BTC timetable, as there are other transport companies in Berland.

If the arrival train and the departure train during a transfer use different platforms of the same station, then passengers are taken by an internal shuttle bus so that they do not have to leave the transfer area. In the case of a transfer at the same platform, the transfer is made on foot through the platform area.

A well-known travel blogger, Iris, plans to travel on BTC trains from city s to city f. Iris dislikes shuttle buses, and since all train rides in this journey are sponsored by BTC, she first wants to minimize the number of inter-platform shuttle trips. Only if there are multiple options with the minimum number of shuttle trips, she plans to minimize the number of train rides.

Given the BTC timetable, as well as the starting and ending cities of the journey, determine the minimum number of times Iris will have to take an internal shuttle bus and the minimum number of train rides she must make to get from the starting city to the destination. In the departure and arrival cities, Iris can choose any platform as she travels by taxi.

Input

The first line of input contains four integers n, m, s, and f ($2 \le n \le 10^5$, $n-1 \le m \le 2 \cdot 10^5$, $1 \le s, f \le n$, $s \ne f$) — the number of cities and the number of train lines in the ICJ timetable, as well as Iris's starting and ending cities, respectively.

Each of the following m lines contains a description of one bidirectional train line and consists of four integers a_i, ta_i, b_i, tb_i ($1 \le a_i, b_i \le n$, $1 \le ta_i, tb_i \le m$, $a_i \ne b_i$) — the first city number, platform number of the first city's station, the second city number and platform number of the second city's station for that line, respectively.

You may assume that it is possible to travel between any two cities using BTC and that any two cities are directly connected by no more than one BTC train line.

Output

Print two integers — the minimum number of internal shuttle bus trips Iris will have to take during her journey and the minimum number of train rides that can be made with that number of shuttle trips.

Example

standard input	standard output
3 3 3 1 1 1 2 1 3 1 2 2 1 2 3 3	0 1

Problem J. Just A Friendly Trick

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	1024 mebibytes

Polina bought n artificial nails, used one of her n nail polishes of n different colors on each of them (she can use the same polish on multiple nails or not use some polish at all), and laid them in a row on the table to dry. While she was away, Vanya came in and decided to rearrange the first k nails from the left, that is, choose a permutation $p_1, \ldots p_k$ and simultaneously move the i-th nail from the left to the p_i-th position for every i from 1 to k.

Vanya doesn't want Polina to notice anything weird, so he wants the sequence of the nail colors on the table to remain unchanged. Polina is very observant and might notice that something's off even with the same order of colors, so, just to be safe, Vanya wants to additionally have the number of cycles in the permutation he chose to be one of Polina's m favorite numbers.

Now Vanya wonders, for each choice of k from 1 to n, how many different permutations satisfying both conditions can he choose. Help him with that! Since the answers can be very large, output them modulo 998 244 353.

Input

The first line contains an integer n ($1 \le n \le 2 \cdot 10^5$): the number of nails on the table.

The next line contains n integers $col_1, \ldots, col_n$ ($1 \le col_i \le n$): colors of nail polishes used on them.

The next line contains an integer m ($1 \le m \le n$): the number of Polina's favorite numbers.

The last line contains m distinct integers $x_1, \ldots, x_m$ ($1 \le x_i \le n$, $x_i \ne x_j$ if $i \ne j$): Polina's favorite numbers.

Output

Print a single line with n integers: answers for each k from 1 to n.

Examples

standard input	*standard output*
3 1 1 2 2 1 3	1 1 1
6 1 1 3 2 6 6 4 6 3 4 2	0 1 2 2 1 2
10 10 2 2 10 10 10 2 10 10 10 5 1 3 9 7 2	1 1 2 3 7 23 53 233 1281 8454

Problem K. Keyword and Numeral

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

You are given an expression of the form $a = b + c$, where a is a keyword (a non-empty string of digits and lowercase or uppercase English letters, starting with a letter), and b and c can be either a keyword or a positive integer numeral in decimal notation (a non-empty string of digits from 0 to 9, starting with a non-zero digit).

You need to place apostrophes in all numerals in the expression, if they are present. The first apostrophe is placed between the third rightmost and fourth rightmost digits (if both digits are present), the second between the sixth rightmost and seventh rightmost (if both digits are present), in general, the k-th apostrophe is placed between the $3k$-th rightmost and $3k + 1$-th rightmost digits (if both digits are present).

Input

The input consists of a single line of the form a=b+c, where a is a non-empty string composed of lowercase and uppercase English letters and digits, starting with a letter, and each of b and c is either a non-empty string composed of lowercase and uppercase Latin letters and digits, starting with a letter, or a non-empty string of digits starting with a non-zero digit. The length of each of the strings a, b, and c does not exceed 1000 characters. There are no spaces between the strings and the characters `+` and `=`.

Output

Output the expression obtained from the original by inserting apostrophes into the numerals. No other changes (such as inserting spaces) are allowed.

Examples

standard input	*standard output*
`ledger2026=cityTax+987654321012345`	`ledger2026=cityTax+987’654’321’012’345`
`Festival2027=Booth42+Prize314159`	`Festival2027=Booth42+Prize314159`

Problem L. Long and Random

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

There is an array a of length n consisting of independent uniformly random integers a_i ($1 \le a_i \le 10^9$). Also, there is an array b of length n consisting of independent uniformly random integers b_i ($0 \le b_i \le 1$).

Laura wants to erase some (possibly zero) elements from array a, then take the prefix of array b with the matching length, and maximize the resulting dot product of the arrays (i.e. $\sum_{i=1}^{m} a_i \cdot b_i$). Help her to do that.

Input

In the first line, there is one integer n ($1 \le n \le 4 \cdot 10^5$)— the length of the arrays a and b.

In the second line, there are n integers $a_1, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the elements of the array a.

In the third line, there are n integers $b_1, \ldots, b_n$ ($0 \le b_i \le 1$) — the elements of the array b.

It is guaranteed that in all tests, except for the first one (from the examples), all numbers a_i and b_i are generated independently from a uniform distribution over the corresponding ranges.

It is guaranteed that there are no more than 20 tests in total.

Output

Print one number — the maximum possible dot product after erasing some elements from array a.

Examples

standard input	*standard output*
10 4 23 7 18 5 16 2 29 11 13 1 0 1 0 1 0 0 1 0 0	70
4 927461835 184739206 603918572 756204391 1 1 1 0	2287584798

Note

In the first example, we can erase the first, fifth and sixth elements from a. The result will be equal to the dot product of the arrays $[4, 6, 5, 3, 6]$ and $[1, 0, 1, 0, 1]$ which equals $4 \cdot 1 + 6 \cdot 0 + 5 \cdot 1 + 3 \cdot 0 + 6 \cdot 1 = 15$.

Problem M. Math, Nero and Seneca

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

Ancient Roman philosopher Seneca was the tutor of the future emperor Nero. Nero once asked him how great a number may be. Seneca answered that there are numbers so large that even the wisest citizens of Rome cannot imagine them; for example, numbers that can be created by multiplying usual Roman numbers greater than 1. To avoid the possible ambiguity, Nero decided that only the integers that have a unique representation in this form are counted.

Now a modern artist decided to create an installation dedicated to Nero and Seneca and wants to use some positive integers not exceeding 10^{18} in it. But he wants to be sure that they can be uniquely represented as a product of one or several numbers greater than 1 in Roman numeral representation (more formally, the representations are counted as one representation if they are identical as the multisets, so the representations $4 \cdot 2$ and $2 \cdot 4$ are the same representation).

Any positive integer strictly less than 4000, and only them, has a unique representation in Roman numerals; other rules for writing Roman numerals are well known to all participants, so they will not be provided here for brevity.

Input

The first line of the input contains one integer t ($1 \le t \le 2 \cdot 10^5$) — the number of the testcases.

Each of the following t lines describes one testcase and contains one integer n ($2 \le n \le 10^{18}$) that needs to be checked.

Output

For each request, print 1 on a separate line if the number can be uniquely represented as a product of Roman numerals greater than 1, and 0 otherwise.

Example

standard input	standard output
4 2024 2025 2026 2027	0 0 0 1