Problem A. Awesome Numbers

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

As it is known, prime numbers play a special role in cryptography. James Bond suggested using prime numbers that can be represented as the difference of the squares of two prime numbers. He called such numbers "awesome".

Your task is to find all awesome numbers in the interval between l and r inclusive. Since the cypher is expected to change with each communication session, you are required to answer several such queries.

Input

The first line of input contains one integer q — the number of queries ($1 \le q \le 10^4$).

The i-th of the following q lines contains two integers l_i and r_i inclusive ($1 \le l_i \le r_i \le 10^{12}$) — the left and the right boundary for the current query.

Output

For each query, print one integer — the number of the awesome numbers between l_i and r_i inclusive.

Example

standard input	standard output
2 1 10 2024 2025	1 0

Problem B. Buildings And Fishermen

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

...In a small settlement on the shores of the Baltic Sea, where the fishermen live, there are two streets parallel to the shorelint. The buildings are numbered with integers from 1 to n such that at each street, each number appears only once.

However, it happens that the order of the building numbers in each line is chosen arbitrarily, meaning that the numbers do not necessarily increase from left to right: the i-th building from the left in the first street has the number p_i, and in the second street, it has the number q_i. Only the buildings from the first line are visible from the sea.

But there is some trouble: the fishermen have some superstitions about the "good" and "bad" numbers, so if they do not see a building with the number x from the sea, their morale changes by a_x; if they see a building with the number x exactly once, it changes by b_x; if they see a building with the number x twice, it changes by c_x.

So they decided to choose some values of i and swap the numbers on i-th buildings at the first and at the second streets (so p_i become old q_i, and q_i become old p_i); the property that the numbers of buildings on each street are unique may be broken, but this is not important for fishermen. But it is important to choose the set of indices in a way such that it maximize the fishermen's morale.

Help the fishermens to solve this problem.

Input

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

The first line of each test case contains a single integer n — the number of buildings in each street.

The second line contains n pairwise distinct integers p_i, where the i-th integer denotes the original number of the i-th building from the left in the first street ($1 \le p_i \le n$).

The third line contains n pairwise distinct integers q_i, where the i-th integer denotes the original number of the i-th building from the left in the second line ($1 \le q_i \le n$).

Lines four to six describe the fishermen superstitions: the fourth line contains n integers a_i, where the i-th integer denotes the change in morale of the fisherman if he did not see the building with number i from the sea while passing by the settlement; the fifth line contains n integers b_i, where the i-th integer denotes the change in morale of the fisherman if he saw the house with number i from the sea exactly once while passing by the settlement; and the sixth line contains n integers c_i, where the i-th integer denotes the change in morale of the fisherman if he saw the house with number i from the sea twice while passing by the settlement ($-10^9 \le a_i, b_i, c_i \le 10^9$).

It is guaranteed that the sum of n across all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output a single integer on a separate line — the maximum change in fishermen morale with the best choice of indices i for which the house numbers between the streets will be swapped.

Example

standard input	standard output
2 3 1 2 3 1 2 3 1 1 1 1 1 1 1 1 1 3 1 2 3 2 3 1 1 2 3 3 2 1 2 1 3	3 8

Problem C. City and Parade

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

There are n squares in the city and m streets connecting them in such a way that there is a route between any two squares via a chain of one or more streets. Additionally, for each street the width of this street is known.

Out of these n squares, k are used for the big military parade.

There are q orders for moving vehicles between the squares involved in the parade. Each order specifies the width of the vehicle that will be moving. For each order, our task is to answer, how many pairs of different squares can be connected by the path, consisting of the streets with a width not less than the width of the vehicle.

Input

The first line of input contains three integers n, m, and k ($1 \le n \le 3 \cdot 10^5$, $0 \le m \le 5 \cdot 10^5$, and $1 \le k \le n$) — the total number of squares and streets in the city and the number of squares involved in the parade, respectively.

The second line contains k pairwise distinct integers from 1 to n — the numbers of the squares involved in the parade.

The i-th of the following m lines contains three integers u_i, v_i, w_i — the squares connected by the i-th street and the width of the i-th street, respectively ($1 \le u_i, v_i \le n$, $u_i \ne v_i$, $1 \le w_i \le 10^9$). Note that there may be two different streets connecting the same pair of squares.

Then follows a line containing a single integer q — the number of orders ($1 \le q \le 5 \cdot 10^5$).

Each of the following q lines contains a single integer d_i — the width of the vehicle to be moved according to the i-th order ($1 \le d_i \le 10^9$).

Output

For each order, output a single integer — the number of pairs of squares involved in the parade between which the vehicle of the specified width can pass.

Example

standard input	standard output
8 9 4 1 8 4 2 1 2 8 2 3 2 1 3 1 1 4 2 4 5 6 5 6 7 5 7 3 7 8 4 8 6 5 4 8 5 2 9	1 2 6 0

Problem D. Debt Calculation

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

The bank "La Bean" provides a credit program for n hours. At the i-th hour, the client receives the loan of size a_i. At the beginning of each hour of the program, except for the first one, the current debt of the client is multiplied by the coefficient b_{i-1}, meaning that if the client started participating from the first hour, then in the middle of the first hour, their debt equals a_1, at the beginning of the second hour it equals $a_1 \cdot b_1$, at the end of the second hour it equals $a_1 \cdot b_1 + a_2$, and so on.

The client can choose the continuous interval of the hours during which they will join the program, meaning they will start taking money from hour l (it is assumed that the debt was 0 for all previous moments), and will take money for the last time at the hour r, after which their debt is fixed at that moment (i.e., it no longer changes).

Your task is to process q scenarios with different l and r and calculate the debt at the end of each scenario. Since the answer may be very large, output the remainder of the division by 998 244 353.

Input

The first line of the input contains one integer $1 \le n \le 10^5$ — the number of hours during which the program operates.

The second line contains n integers $1 \le a_i \le 10^9$. The i-th of these numbers specifies the size of the loan issued at the i-th hour.

The third line contains $n - 1$ integers $1 \le b_i \le 10^8$ — the coefficients by which the debt is multiplied between the i-th and $i + 1$-th hours.

The next line contains a number $1 \le q \le 10^5$ — the number of scenarios. Following this are q lines of queries. Each of these lines contains two numbers l_i, r_i $(1 \le l_i \le r_i \le n)$ — the first and last hour of the client's participation in the credit program according to the given scenario.

Output

For each query, output a single integer on a separate line — the remainder of the division of the debt amount at the end of the r_i-th hour by 998 244 353.

Example

standard input	standard output
5 1 2 3 4 5 2 2 4 2 4 1 1 1 2 2 5 3 4	1 4 69 16

Problem E. Extremelly Thin Backpack

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

Given the thin backpack of volume V and set of items, for each item two parameters are known:

- c — the value of the item.
- v — the volume of the item.

Since the backpack is extremelly thin, the following condition holds for any item: $1 \le v \le V$ and $3 \cdot v + 100 \ge V$, where v is the volume of the item.

What is the maximum total value of the items that can fit into the backpack?

Input

The first line of the input contains one integer t — the number of test cases ($1 \le t \le 10$).

The following lines contain the test cases. The first line of each test case contains two integers n — the number of the items and V — the capacity of the backpack ($1 \le n \le 10^5$, $1 \le V \le 10^9$).

The next line contains n integers v_i — the volumes of the i-th item, in order. It is guaranteed that $1 \le v_i \le V$, and that $3 \cdot v_i + 100 \ge V$.

The following line contains n integers c_i — the values of the items, in order ($1 \le c_i \le 10^9$).

Output

For each test case, output one number — the maximum total value of the items that can be carried out in the backpack.

Example

standard input	standard output
2 4 12 4 6 9 5 3 4 7 4 2 10 9 9 1 2	8 2

Problem F. From One Isotope To Another

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

This is an interactive problem

In the laboratory, there are n radioactive isotopes numbered from 1 to n. The radioactivity of isotope i is equal to i.

The following events occur:

- Opening $(1\ x\ y)$: a reaction is opened in the laboratory that transforms isotope x into isotope y and vice versa.
- Query $(2\ x\ r)$: — the boss asks the question: "how many isotopes can be obtained from isotope x through some (possibly zero) number of reactions, such that the maximum radioactivity during the process does not exceed r"?. It is guaranteed that $r \geq x$.

Since the laboratory staff is quite disciplined, upon receiving a query, they must answer it and only then continue their work.

Write a program that simulates the process.

Interaction Protocol

The interaction begins with the jury program outputting two integers n and q $(1 \leq n, q \leq 5 \cdot 10^5)$ — the number of isotopes and the number of events, respectively.

Next, the jury program outputs the events. In the case of an opening, the jury program outputs three integers $1\ x\ y$, indicating that isotope y can be obtained from isotope x and vice versa $(1 \leq x, y \leq n$, $x \neq y)$. The participant's program should not output anything in response to the first type of event (opening).

If the boss asks a question, the jury program outputs three integers $2\ x\ r$, where x is the starting isotope, and r is the maximum radioactivity value that can be reached during the transformation process $(1 \leq x, r \leq n, x \leq r)$. In response to this event, the participant's program should output a single integer — the number of isotopes that can be obtained from the starting isotope while adhering to the requirements of the problem statement. After outputting the answer, do not forget to output a newline character and flush the input-output buffer using the function `flush` of your chosen programming language.

Example

standard input	standard output
6 10 1 4 5 2 1 5 1 6 1 2 6 6 1 4 6 1 5 4 1 3 1 2 3 5 2 5 5 2 5 6	1 2 2 2 5

Problem G. GymForces Show

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

The sport show features n athletes from the GymForces, each with a rating a_i on GymForces (two athletes may have the same rating).

Let's define the beauty of the formation as follows: we select an athlete, take the minimum rating among all athletes standing no further right than the selected one, and the maximum rating among all athletes standing no further left than the selected one, and multiply them. To obtain the final beauty, we will sum the resulting products for all n athletes.

More formally, beauty is defined as $\sum_{i=1}^{n} \min(a_1, a_2, ..., a_i) \cdot \max(a_i, a_{i+1}, ..., a_n)$.

During the show, the number of participating athletes decreases in the following way: there are a total of q rearrangements. In j of them, an athlete standing in position b_j from the left among those who are still in the show leaves.

Your task is to determine the initial beauty of the formation and the beauty of the formation after each of the rearrangements. All actions (determining the next departing athlete and calculating beauty) are performed only with the athletes who are still present in the show.

Input

In the first line, you are given the numbers $1 \le n \le 2 \cdot 10^5$ – the initial number of athletes in the show and $0 \le q \le n - 1$ – the number of rearrangements. The second line contains n numbers – the ratings of the athletes a_i ($1 \le a_i \le 10^6$), where the i-th number specifies the rating of the i-th athlete in the initial formation.

The third line contains q numbers b_j; the j-th of them specifies the position of the athlete leaving the show during the j-th rearrangement. The position is defined as the number of the position from the left among the remaining athletes at the moment of the j-th rearrangement; it is guaranteed that at least b_j athletes are still remaining in the show by that time.

Output

Output $q + 1$ numbers – the values of the beauty of the formation at the beginning of the show and after each of the rearrangements.

Example

standard input	standard output
6 3 8 5 7 4 2 9 3 5 2	234 189 109 84

Problem H. Hall 1666

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

This is an interactive problem

At the Intergalactic Collegiate Programming Contest (ICPC) the problem analysis was held in the hall 1666. The contest is held in the ancient castle, so the halls were enumerated by Roman numerals. But this morning instead the Roman numerals there were signs with lowercase Latin letters on the doors.

It turned out that during the night the mad interior designer replaced the "out-of-date" Roman numerals (the uppercase Latin letters `I`, `V`, `X`, `L`, `C`, `D`, and `M`) with lowercase letters from `a` to `g` in some order, but the property that identical Roman numerals were replaced with identical letters, and different ones with different letters is held.

It is not surprising that a huge queue of contestants and volunteers gathered at the lobby, eager to find out the hall numbers. The problem was further complicated by the fact that the head of security, who had information about the new numbering from the designer himself, could only answer questions "does a hall with a certain number exist", where the number is a string of letters from `a` to `g`. Moreover, according to security's personal sense of the maximum allowable intensity of his own working day, one visitor could ask no more than 16 questions...

There are 3999 halls in the castle, which are numbered with Roman numerals according to the following rules for writing Roman numbers.

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 reverse 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 reverse notation is used.
- A number containing several 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 this place in the Roman representation.
- The largest number that can be represented in the Roman numeral system is 3,999 (MMMCMXCIX).

So your task is to find the text that is written now on the door of hall 1666, using no more than 16 questions.

Interaction Protocol

The interaction begins with the jury program outputting a single integer t — the number of scenarios ($1 \le t \le 125$).

In each scenario, the interaction starts with the participant's program outputting a query in the form of `?` s, where s is a non-empty string of length no more than 15 characters, consisting of lowercase Latin

letters from `a` to `g`. In response, the jury program will output 1 if, after replacing the letters with the corresponding Roman numerals, a valid Roman numeral is obtained, and 0 otherwise. It is guaranteed that the correspondence of letters to Roman numerals is determined before the start of each scenario and does not change during the scenario (i.e., the interactor is not adaptive). You may ask no more than 16 queries.

If your program is ready to output the answer — the encoding of room number 1666 (MDCLXVI in Roman), output the answer in the form `!` *ans*, where *ans* is a permutation of the first 7 letters of the Latin alphabet that encodes the number 1666. Outputting the answer does not count as a query.

After that, the program should proceed to execute the next scenario or terminate if this scenario was the last one.

Example

standard input	standard output
1 1 0 0 1	? gce ? ace ? gege ? fffgdddbccceaaa ! fgbecda

Note

Do not forget to output a newline character after each query or response and flush the input-output buffer using the `flush` function of the programming language used.

Problem I. Interplanetary Business

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

Alice and Basilio jointly purchased land at the Moon. Alice invested k times more than Basilio.

The acquired land is a simple polygon, which is not necessarily convex, defined by its n vertices listed in clockwise or counterclockwise order.

Now they want to divide the acquired territory according to each person's contribution by building a fence in the meridional direction (i.e., parallel to the y-axis) so that the area of the territory to the left of the fence (which will go to Basilio) to the area of the territory to the right of the fence (which will go to Alice) is in the ratio of 1 to k.

Input

The first line of input contains two integers n ($3 \le n \le 10^5$) and k ($1 \le k \le 1000$). Each of the following n lines contains two integers x_i and y_i — the coordinates of the i-th vertex of the purchased land in the chosen order of traversal.

It is guaranteed that two sides of the polygon bounding the territory intersect if and only if they share a vertex, and that the area of the territory is not zero.

All coordinates do not exceed 10^5 in absolute value.

Output

Output a single real number — the x coordinate of the vertical fence with an absolute or relative error no worse than 10^{-6}.

Examples

standard input	standard output
10 1 0 0 2 2 1 3 1 4 2 6 0 7 -2 6 -1 4 -1 3 -2 2	0
4 2 -1 -1 -1 11 14 11 14 -1	4.000000

Problem J. Jackpot

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

A new game called "triangle" had recently appeared in the casino. The deck for this game consists of n cards, on which the numbers a_1, a_2, ..., a_n are written.

The player divides the cards into three non-empty piles so that each card goes into exactly one pile, after which the dealer draws one card from the first, second, and third piles. If among them there is a number that is not less than the sum of the other two, the dealer wins. If this did not happen after the several actions of the dealer, the player wins the jackpot. The experienced player immediately noticed the markings on the backs of the cards, by which the dealer could know the value of each card. The dealer obviously was cheating...

But the player still can earn the jackpot, if it is possible to divide the cards into piles in such a way that no matter how the dealer chose, the player would always win.

Preparing for the series of the games, the player wants to know the number of different ways to divide the deck into three parts so that the dealer always loses. Two options are considered different if there exists a pair of cards (not necessarily with different values on them) such that in one option these cards are in the same group, while in the other they are in different groups. Since the answer can be very large, output the remainder of its division by 998 244 353.

Input

The first line contains one integer n — the number of cards in the deck ($3 \le n \le 3000$).

The second line contains n numbers a_i — the values of the cards in the deck ($1 \le a_i \le 10^9$).

Output

Output the remainder of the division by 998 244 353 of the number of ways to divide the deck of cards into three groups so that the dealer always loses, and the player always earns the jackpot.

Examples

standard input	standard output
4 3 3 5 6	3
3 1 1 1	1
3 1 2 3	0

Problem K. King Primes

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

James Bond finally found that the cipher based on the awesome numbers from the task A is cryptographically weak, and he decided to propose a new one.

James supposed that the reliability of the cipher would increase if he used prime numbers that can be expressed as the sum of the squares of two prime numbers. He called such numbers "King Primes" after His Majesty Charles III.

Your task is to find all the King Primes in the interval between l and r inclusive. Since the set of prime numbers is expected to change with each communication session, you are required to answer several such queries.

Input

The first line of input contains one integer q — the number of queries ($1 \le q \le 10^6$).

The i-th of the following q lines contains two integers l_i and r_i inclusive ($1 \le l_i \le r_i \le 10^{12}$) — the boundaries for the next query.

Output

For each query, output one integer — the number of the King Primes between l_i and r_i inclusive.

Example

standard input	standard output
2 9 29 20 24	2 0

Problem L. Loose Connection

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

This is a double-run problem

You need to transmit 8 bytes of information per one session of radio communication. You are allowed to transmit exactly 9 bytes, but, due to the magnetic storm, the communication channel is unstable, and some bytes (no more than four) may be transmitted in inverted form, meaning all zeros are replaced with ones and vice versa (for example, 10010010 is replaced with 01101101).

Your task is to write a program that ensures the reception and transmission of messages. Your program will be run independently two times: during the first run, it receives 8 bytes in the form of a 16-digit hexadecimal number (possibly with leading zeros) that needs to be forwarded, after which it generates an 18-digit hexadecimal number — the message to be transmitted. During the second run, it receives 9 bytes (with possible distortions) and must restore the original 8 bytes.

Input

The first line of the input contains an integer n — the number of messages ($1 \le n \le 10^5$).

If the program is running in transmission mode, the second line of the input contains the word `encrypt`. The following n lines each consist of exactly 16 hexadecimal digits (`0-9`, `A-F`) and represent the 16-digit hexadecimal number that needs to be transmitted.

If the program is running in reception mode, the second line of the input contains the word `decrypt`. The following n lines each consist of exactly 18 hexadecimal digits and represent the 18-digit hexadecimal number with distortions.

Output

In transmission mode, output for each message in the order they appear in the input file a hexadecimal string of 18 characters — 9 bytes to be transmitted.

In reception mode, output for each received nine-byte string in the order they appear in the input file a hexadecimal string of 16 characters — the corresponding original message.

Examples

standard input	standard output
1 encrypt 0123DEADBEEF1337	C020ADDED2FEED4ACE
1 decrypt C0DF52DED2FEED4ACE	0123DEADBEEF1337

Problem M. Modifying The Queue

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

The queue has a fixed length of n, and the positions in the queue are numbered from head to tail with the indices from 1 to n. There are n positive integers a_i in the queue (the integers in the queue may repeat).

With one operation you can either remove any even number located at an odd position or remove any odd number located at an even position. After the removal the numbers that were further back in the queue shift forward by one position (thus the queue shortens but remains continuous).

Determine the maximum number of the operations you can perform.

Input

In the first line, you are given the number of elements in the sequence $1 \le n \le 10^5$. In the next line, the elements of the array are given – n positive integers $1 \le a_i \le 10^9$ in the order of their positions in the queue from head to tail.

Output

Output the maximum number of operations you can perform.

Example

standard input	standard output
3 5 1 2	2