Problem B. Bridge

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

It's a sunny day with good scenery, and you come to the park for a walk. You feel curious that there are many old guys gathering by a bridge, and want to take a look at what happened.

There are exactly n old guys on each side of the bridge, and they all want to go across the bridge, take some time for relaxing on the other side, and finally go across the bridge back to the original side. However, they are too old to cross the bridge by themselves.

Driven by the golden spirit in your heart, you want to help these $2n$ old guys. Initially, you are on one side of the bridge. It takes t minutes for you to go across the bridge and x minutes for an old guy relaxing. You may help an old guy when you cross the bridge, which doesn't take extra time.

As a master of time management, you want to know the minimum time needed to help all these $2n$ old guys. Please write a program to calculate this minimum time.

Input

The first line contains one integer T ($1 \le t \le 10^4$), indicating the total number of test cases.

For each of the next T lines, there are three integers n, x, t ($1 \le n, x, t \le 10^9$), as explained in problem statement.

Output

You should output exactly T lines. Each line should contain exactly one integer, indicating the minimum time you needed in each test case.

Example

standard input	standard output
3 2 2 2 3 1 10 11 45 14	16 120 616

Note

For the first case of the sample data, the optimal plan is shown below, where a numerical digit denotes an old guy, `|` denotes the bridge, and `x` denotes you.

```
Time
  0     x 1 2 | 3 4
  2         2 | 3 4 1 x
  4     x 3 2 | 4 1
  6         3 | 4 1 2 x
  8     x 4 3 | 1 2 x
 10         4 | 1 2 3 x
 12     x 1 4 | 2 3
 14         1 | 2 3 4 x
 16     x 2 1 | 3 4
```

Problem C. Combine The Gears

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

With the rapid development of society, the demand for high-precision clocks is constantly rising. Recently, the China Clock Production Company is developing a new type of clock, which can represent a wide range of times.

The novel clock displays the current time in an unusual fashion. The clock consists of several pointers, each controlled by a gear. All gears rotate synchronously – one tooth per period. However, the numbers of teeth of the gears may differ. If a gear has t teeth, then the corresponding pointer can point to t different directions, denoted $0, 1, 2, \cdots, t-1$, respectively, where 0 is the initial direction. Furthermore, if a clock is equipped with n pointers, the i-th of which is controlled by a t_i-tooth gear, then the i-th pointer will point to $k \bmod t_i$ after k periods of time.

The price for a t-tooth gear is t yuan. Given a total budget of b yuan, you need to design a combination of gears, such that the number of valid combinations of directions of pointers is maximized, and the total cost on gears does not exceed the budget. A combination of directions $(d_1, d_2, \cdots, d_n)$ is valid, if it can be written

$$(k \bmod t_1, k \bmod t_2, \cdots, k \bmod t_n)$$

for some nonnegative integer k, where t_i is the number of teeth of the i-th gear. Since the answer may be too large, output the answer in natural logarithm (logarithm with base $e = 2.718281828\cdots$).

Input

The first line of input is a single integer T ($1 \le T \le 30\ 000$), indicating the number of test cases. Each test case is a single line of an integer b ($1 \le b \le 30\ 000$), denoting the total budget.

Output

For each test case, print the natural logarithm, within an absolute or relative error of no more than 10^{-6}, of the maximum number of valid combinations, in a single line.

Example

standard input	standard output
3 2 7 10	0.693147181 2.484906650 3.401197382

Note

For the second sample data, a 3-tooth gear along with a 4-tooth gear may yield 12 different combinations of directions, with total cost exactly being 7. So you should print the value of $\ln 12$, which is approximately 2.484906650.

Problem E. Evil Problemsetters

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	1024 mebibytes

Due to the challenging problems, some of the contestants decide to escape from this contest. However, to prevent this from happening, the EVIL problem setters made a labyrinth at the stadium's exit. The labyrinth is made of an $n \times m$ grid, on which lie the entrance and the exit, and k black holes. Contestants who accidentally step into any black hole will fall into it and thus can never escape from the contest.

What's worse, the problem setters may also adjust the coordinates of the entrance and the exit. You, a poor contestant, who start from the entrance and wish to reach the exit without stepping into any of the black holes, can only move to one of the four adjacent cells in each step. You want to know, after each time the problem setters change the coordinates of the entrance and the exit, what's the minimum number of steps needed to reach the exit starting from the entrance?

Input

The first line of the input contains four integers n, m, k, q ($1 \le n, m \le 200\,000, nm \le 200\,000, 0 \le k \le 42$, $1 \le q \le 100\,000$), denoting the number of rows, the number of columns, the number of black holes in the labyrinth, and the number of queries, respectively.

The following k lines contain the description of the black holes. Each of these lines contains two integers x, y ($1 \le x \le n, 1 \le y \le m$), denoting the coordinates of a black hole. No two black holes are located at the same position.

The last q lines contain the description of the queries. Each of the q lines contains four integers x_s, y_s, x_t, y_t ($1 \le x_s, x_t \le n, 1 \le y_s, y_t \le m$), where (x_s, y_s) is the coordinates of the entrace and (x_t, y_t) is the exit.

Output

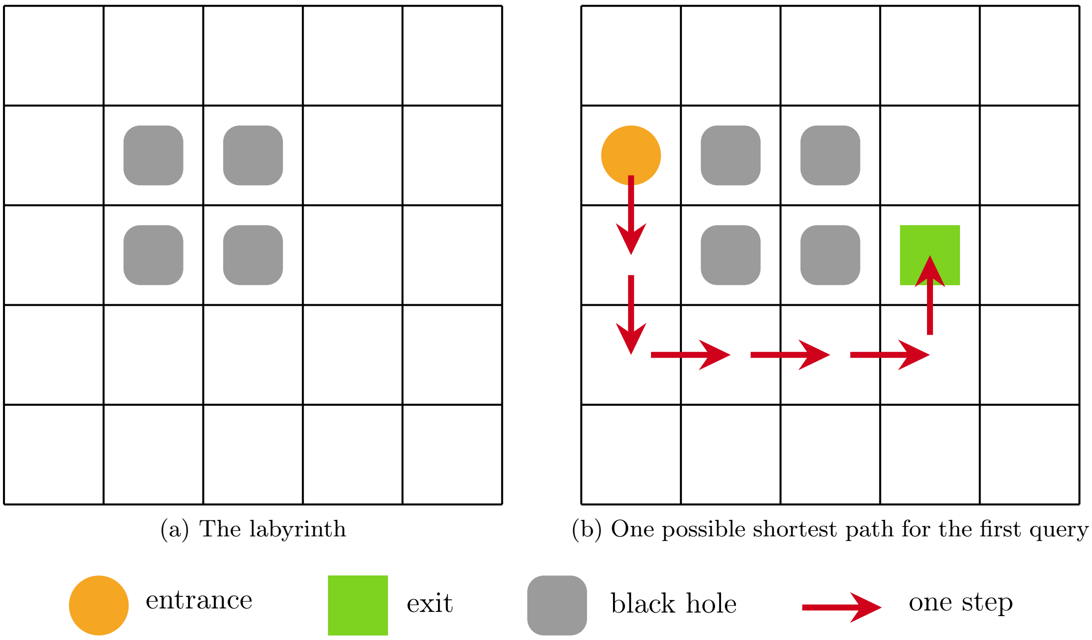
For each query, output a number in a line, denoting the minimum number of steps needed to reach the exit starting from the entrance. If it is impossible to reach the exit, output `-1` instead. It should be considered impossible when the entrace or the exit coincides with a black hole.

Examples

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>5 5 4 7
2 2
2 3
3 2
3 3
2 1 3 4
1 1 1 1
2 2 2 2
1 1 1 5
2 2 5 5
2 1 2 4
1 1 3 3</pre></td><td><pre>6
0
-1
4
-1
5
-1</pre></td></tr>
<tr><td><pre>2 3 2 1
1 2
2 1
1 1 2 3</pre></td><td><pre>-1</pre></td></tr>
</table>

Note

The plots for the labyrinth and the first query of the first sample data are shown below.



(a) The labyrinth

(b) One possible shortest path for the first query

Рис. 1: Plots for sample test data

Problem G. Game With Stones

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Have you ever heard of the Nim game? Probably your answer is yes, but I bet you've never heard a new game called "Steins;Game"! The game is named so because it is a game played with many piles of stones, and *Stein* means *stone* in German.

Alice and Bob want to play this game now. The game is played as follows:

> There are n piles of stones arranged in a row. Each pile of stones is colored either white or black. The i-th pile consists of a_i stones. Starting with Alice, the two players take turns to choose to do one of the following things
>
> - take any positive number of stones from the smallest black pile (i.e., the smallest one among all black piles); or
> - take any positive number of stones from any white pile.
>
> The game ends when no stones are remaining, and the player who takes the next move loses the game.

Now that all piles of stones are fixed, but not yet colored. Bob has bribed the referee to get the chance to paint all piles of stones by himself. Now he wonders how many ways to paint the piles of stones so that he can win the game, assuming Alice and Bob play optimally? Since the answer may be too large, you only need to output it modulo 1 000 000 007.

Input

The first line of the input contains one integer n ($1 \le n \le 10^5$), denoting the number of piles in the game.

The second line contains n integers $a_1, a_2, \cdots, a_n$ ($1 \le a_i \le 10^{18}$), denoting the number of stones in each pile.

Output

Print one integer in a line, denoting the number of ways of painting the stone piles for Bob to win the game, modulo 1 000 000 007.

Examples

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

Note

For the first test case, Bob can win under any of the four ways of paintings.

For the second test case, the only way for Bob to win is to paint both two piles to black, so that Alice is forced to take the first pile in the first move, and then Bob can take the whole second pile and win the game.

For the third test case, no matter how Bob paints the only pile, Alice can take the whole pile and win the game.

Problem H. Hotels

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Located at the easternmost tip of Shandon Peninsula, Weihai is one of the most famous tourist destinations all over China. There are beautiful hills, seas, bays, springs, islands, and beautiful beaches in Weihai. It is also a coastal city abundant in seafood, including prawn, sea cucumber, abalone, shellfish, and algae.

Attracted by the distinctive scenery and pleasant environment, three theoretical computer scientists plan to have a trip to Weihai. However, they cannot reach a consensus on accommodation, since some people prefer some hotels while other people like others. They decide to stay in possibly different hotels at night and meet in one hotel the next day. The hotel they meet may not necessarily be one of the hotels they stay in.

There are some roads connecting the hotels in Weihai. The roads are specially designed such that there is a unique path between every pair of hotels. Every theoretical computer scientist has prepared a list of candidate hotels before their trip starts. When they arrive in Weihai, each of them will uniformly and independently choose one hotel from the candidate hotel list. Also, they will meet in a hotel such that the total length of their routes is minimized. As a member of the theoretical computer science group, can you tell the expected total length of their routes?

Input

The first line of the input contains a single integer n ($1 \le n \le 200\,000$), denoting the number of hotels in Weihai. Then follow $n-1$ lines, describing the roads connecting the hotels. Each of the $n-1$ lines contains three integers u, v, w ($1 \le u, v \le n, u \ne v, 1 \le w \le 1000$), denoting a road of length w connecting the hotels numbered u and v. It is guaranteed that there is a unique path between every pair of hotels.

The last three lines of the input specify the candidate hotel lists, one for each theoretical computer scientist. Each line begins with a single integer m ($1 \le m \le n$) and m distinct integers $a_1, a_2, \cdots, a_m$ ($1 \le a_i \le n$), meaning that the candidate hotel list contains the hotels numbered $a_1, a_2, \cdots, a_m$.

Output

Print the expected total length of their routes within an absolute or relative error of no more than 10^{-6}.

Examples

standard input	standard output
3 1 2 1 2 3 2 1 1 1 2 1 3	3
5 1 2 3 1 3 5 2 4 7 2 5 11 3 2 4 5 4 1 2 3 5 2 1 3	13.958333333333

Problem I. Internet Chats

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	256 mebibytes

While we enjoy chatting with friends on the internet, it is always annoying that we are overwhelmed by lots of messages in various chat groups. A great majority of these messages are actually not interesting to us, but we may miss some important notices if we silence these groups. How many messages do we receive from all online chat groups? Nobody has ever seriously gone into this question.

As an assistant researcher in the school of informatics, you are required to investigate the number of online messages we receive every day. We have already sampled n groups and m students. Every group contains a subset of the m students, which is possibly empty. Also, the members of the groups are constantly evolving; old members may quit, and new members may join in a chat group. Members can send messages in the group; the message is broadcast to all other members currently in the same group.

Now we have collected the log of these chat groups. The log is a sequence of events, which may be a student joining in a group, quitting a group, or sending a message in a group. Your task is to compute the total number of messages received by every student.

Input

The first line of the input contains three integers n, m, s ($1 \le n \le 100\,000, 1 \le m \le 200\,000, 1 \le s \le 1\,000\,000$), denoting the number of groups, the number of students and the number of events in the log.

The next s lines give the events in the log in chronological order. Each of them contains three integers t, x, y ($t \in \{1, 2, 3\}, 1 \le x \le m, 1 \le y \le n$) specifying an event, which may fall into one of the following three categories:

- If $t = 1$, it means that the x-th student joined in the y-th group. It is guaranteed that the student was not in the group before.
- If $t = 2$, it means that the x-th student quitted the y-th group. It is guaranteed that the student was currently in the group.
- If $t = 3$, it means that the x-th student sent a message in the y-th group. It is guaranteed that the student was in the group now.

Initially, all groups were empty.

Output

Output m lines. The i-th line contains an integer, denoting the total number of messages the i-th student received.

Examples

standard input	standard output
3 3 10 1 3 2 1 3 1 1 1 2 1 2 1 3 1 2 2 3 1 3 3 2 3 2 1 3 3 2 3 2 1	2 0 1
2 5 10 1 1 2 3 1 2 2 1 2 1 3 2 1 1 2 3 1 2 3 3 2 1 4 2 3 3 2 1 5 1	2 0 1 1 0

Problem J. Julius Caesar and Kazusa

Input file:	`standard input`
Output file:	`standard output`
Time limit:	13 seconds
Memory limit:	256 mebibytes

Have you ever heard of the Caesar cipher? It is one of the simplest and best-known encryption techniques. Named after Julius Caesar, he used this cipher to communicate with his generals.

Caesar cipher is a type of substitution cipher in which each letter in the plaintext is *shifted* a certain number of places down the alphabet. The alphabet is considered wrapped around. For example, with a shift of 1 in the Latin alphabet, A would be replaced by B, B would become C, Z would be A, and so on.

Kazusa now has an array a of integers $a_1, a_2, \ldots, a_n$ of length n, with each a_i in the range $[0, 65\,536)$, and she wants to encrypt it using Caesar cipher several times. She selects an interval $[l, r] (1 \le l \le r \le n)$ each time, and makes an encryption using Caesar cipher with shift of 1 to the numbers in the interval. Formally, for all $l \le i \le r$, this transforms a_i into $(a_i + 1) \bmod 65\,536$.

However, while Kazusa is encrypting the array, her sister, Setsuna, raises some questions about the array. Each query asks on the current copy of the array, where zero or more encryptions using Caesar cipher has been done. Each query is given by three integers x, y, L, which asks whether the two strings $a_x, a_{x+1} \ldots a_{x+L-1}$ and $a_y, a_{y+1} \ldots a_{y+L-1}$ are same.

While Kazusa is busying doing the encryption, she has no time to answer these queries. Could you please help her?

Input

The first line contains two integers n, q $(1 \le n, q \le 500\,000)$, denoting the size of the array and the number of operations, respectively. The second line contains n integers $a_1, a_2, \ldots, a_n$ $(0 \le a_i < 65\,536)$, denoting the initial array. The next q lines describe the operations, each operation is in one of the two following types:

- Operation of type 1 contains three integers $1, l, r$ $(1 \le l \le r \le n)$, meaning that Kazusa made an encryption using Caesar cipher with shift 1 to the numbers in the interval $[l, r]$;
- Operation of type 2 contains four integers $2, x, y, L$ $(1 \le x, y \le n, \max\{x, y\} + L - 1 \le n)$, meaning that Setsuna asked a query whether the two strings are the same.

Output

For each operation of type 2, if the two strings are same, please print `yes` in one line; otherwise print `no`.

Examples

standard input	standard output
5 6 1 2 1 2 1 2 1 2 2 2 1 3 3 1 1 1 1 3 5 2 1 2 4 2 1 2 2	no yes no yes
3 3 0 65535 65535 2 1 2 2 1 2 3 2 1 2 2	no yes

Note

The first test case is explained below.

Operation	Array	Description
1	[1,2,1,2,1]	[1,2] and [2,1] are different
2	[1,2,1,2,1]	[1,2,1] and [1,2,1] are same
3	[2,2,1,2,1]	the first element is shifted by one
4	[2,2,2,3,2]	the third to the fifth elements are shifted by one
5	[2,2,2,3,2]	[2,2,2,3] and [2,2,3,2] are different
6	[2,2,2,3,2]	[2,2] and [2,2] are same

Problem L. Light Version Of Famous Task

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	256 mebibytes

The ABC conjecture (also known as the Oesterlé–Masser conjecture) is a famous conjecture in number theory, first proposed by Joseph Oesterlé and David Masser. It is formally stated as follows:

Рис. 2: Shinichi Mochizuki

> For every positive real number ε, there are only finitely many positive integer triples (a, b, c) such that
>
> 1. a and b are relatively prime;
> 2. $a + b = c$; and
> 3. $c > \text{rad}(abc)^{1+\varepsilon}$,
>
> where
> $$\text{rad}(n) = \prod_{\substack{p|n \\ p\in\text{Prime}}} p$$
> is the product of all distinct prime divisors of n.

Shinichi Mochizuki claimed to have proven this conjecture in August 2012. Later, Mochizuki's claimed proof was announced to be published in *Publications of the Research Institute for Mathematical Sciences* (RIMS), a journal of which Mochizuki is the chief editor.

Spike is a great fan of number theory and wanted to prove the ABC conjecture as well. However, due to his inability, he turned to work on a weaker version of the ABC conjecture, which is formally stated as follows:

> Given a positive integer c, determine if there exists positive integers a, b, such that $a + b = c$ and $\text{rad}(abc) < c$.

Note that in the original ABC conjecture, the positive integers a and b are required to be relatively prime. However, as Spike is solving an easier version of the problem, this requirement is removed.

Input

The first line of input contains one integer T ($1 \leq T \leq 10$), the number of test cases.

The next lines contain description of the t test cases. Each test case contains one line, including an integer c ($1 \leq c \leq 10^{18}$).

Output

For each test case, if there exist two positive integers a, b satisfying $a + b = c$ and $\text{rad}(abc) < c$, then output `yes` in a line, otherwise output `no` instead.

Example

standard input	standard output
3 4 18 30	yes yes no

Note

For the first test case, we have $2 + 2 = 4$ and $\mathrm{rad}(2 \times 2 \times 4) = 2 < 4$.

For the second test case, we have $6 + 12 = 18$ and $\mathrm{rad}(6 \times 12 \times 18) = 6 < 18$.

For the third test case, there's no solution.

Problem M. Maximize Minimal Pair Rating

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

The KV Cup is the online programming competition for pairs of programmers.

Coach of some big university is in charge of registration of teams for this competition. There is an even number of students who desire to compete, so each student does compete.

For each student the rating on the famous site Forcedcoders is known. Coach considers rating of the pair as sum of ratings of the contestants in it. Coach wants to form teams in such a way, that lowest rating of the pair is maximized. Help him to find this rating.

Note that each student may participate exactly in one pair.

Input

The first line of input contains a single integer n ($1 \le n \le 10^5$), representing the number of students who desire to enter the KV Cup. It is guaranteed that n is an even number. Each of the following n lines contains a single integer r ($1 \le r \le 10^6$), representing the rating of a student.

Output

Print one integer — maximal possible value of lowest rating of the pair.

Examples

standard input	standard output
4 3010 3020 3030 3050	3050
4 2013 2012 2019 2014	4027

Problem N. Negative People

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Two programmers sit in a car following the bus in a traffic jam. At the stop two people enter the bus. Then, three people are observed leaving the bus. Then one of the programmers said: If one person is to enter now, the bus will be empty!

Those programmers did several observations and wrote down numbers of people who entered and left the bus at the consecutive stops.

Your goal is to write a program that will calculate the minumum number of passenger in the bus before their observations started.

Input

The first line of the input consists of a single integer T ($0 < T \leq 50$) — the number of test cases.

Each of the test cases begins with a line containing a single integer M ($0 < T \leq 50$) — number of observations. Then sequential observations follow each on the new line. Each observation is described by two integers P_1 and P_2 separated by a space — number of people, entering the bus, and then number of people, leaving the bus, respectively ($0 \leq P_1, P_2 \leq 1000$). Note that one observation consists of two events: first, P_1 people enter the bus, then P_2 people leave the bus.

Output

For each test case, print the minimum number of people that would have to have been inside the bus at the beginning.

Example

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

Problem O. Order Backward

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Given a list of words consisting of only capital English letters, create a list by reversing them and printing the reversals in alphabetical order.

Input

Each input will consist of a single test case.

The first line of input will contain an integer n ($1 \le n \le 1,000$) indicating the number of words. On the following n lines will be the words, one per line.

The words will be from 1 to 100 letters long. The words will consist of only capital letters, and there will be no spaces or blank lines.

Output

Output the words, reversed and sorted, one word per line.

Examples

standard input	standard output
3 WEIHAI GRAND PRIX	DNARG IAHIEW XIRP

Problem P. Play The Guessing Game

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

A guessing game is played in the next way: given a goal number N (hidden) and a range (X, Y) containing it (not including X and Y), everyone takes turns to guess what N is. After every wrong guess, the range replaces to a new range with either X or Y replaced by the guessed number. The game ends when a guess hits the number N.

Supposing that everyone always guesses the median of the range (if there are two, choose the smaller one), how many guesses need to be played for the game to end?

Input

The first line of the input file input contains three integers N, X and Y $(0 \leq N, X, Y \leq 10^4)$.

Output

Print number of guesses that will be played.

Examples

standard input	standard output
42 20 80	3