Problem A. Alice and Money Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Alice found a tree with N vertices. Each vertex of the tree is numbered by an integer 1 to N. Also, one or two coins grow at each vertex.

Alice plays the following single-player game on the tree. First, Alice chooses one vertex of the tree and puts a token on it. After that, the following action is repeated:

Alice takes exactly one coin from the vertex on which the token is now, after which selects one vertex adjacent to the given one, in which there are coins left, and moves the token to this vertex. If there is no coin in any adjacent vertex, the game ends.

What is the largest number of coins Alice will collect during the game?

Input

The input data contains no more than 50 test cases.

The first line of a test case contains one integer N — the number of vertices of the tree ($2 \le N \le 10^5$).

The second line contains a string of length N, i-th character is equal to 1, if the i-th vertex contains one coin, and to 2 if i-th vertex contains two coins.

The last $N-1$ lines are describing the tree. i-th of those lines contains one integer p_i ($1 \le p_i \le i$) — the number of the vertex connected with the vertex $i+1$.

The input is terminated with a test case with $N = 0$, which should not be processed.

Output

For each test case in the new line print one integer — maximal number of the coins that Alice can collect while the game.

Example

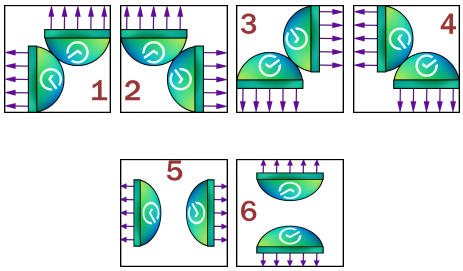
standard input	standard output
2 12 1 5 11212 1 2 3 4 7 2122212 1 1 1 1 1 1 5 21212 1 2 3 4 0	3 5 5 5

Problem B. Warehouses for SberMarket

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Maintaining a large online store that delivers goods across the country, such as SberMarket, requires the organization of huge warehouses. The warehouses are located underground and served by automatic trolleys. In some areas, carts must be lifted to the surface. For this, special in-out gates are installed.

On a 1×1 kilometer section, the following gate installation configurations are allowed:



A new central warehouse is being set up for SberMarket. The warehouse area is a fenced rectangle w for h kilometers. In every cell size 1×1 kilometer **need to** place one of 6 gate configurations.

Purple arrows are drawn from the side where the gate is located.

Automatic carts move strictly in a straight line from one gate to another; in the event of a meeting of two opposite carts, they use the collision avoidance system and continue to move along their straight line.

Automatic carts can leave any gate and enter any gate; also in configurations 5 and 6, the cart can move between the doors vertically for configuration 5 and horizontally for configuration 6.

If the cart hits the gate from the back or crashes into the fence, an accident occurs.

It is required to indicate for each sector of the warehouse which of the 6 door configurations should be located in it so that accidents are impossible.

Input

The first line of the input contains two integers w and h ($1 \le w, h \le 10^3$) — the width and height of the SberMarket warehouse.

Output

Print h lines, each line contains w numbers in the range from 1 to 6, separated by spaces — numbers of gate configurations in the corresponding squares of the warehouse. If there are several answers, print any.

If it is impossible to place the gates without risk of accidents, print -1 instead.

Examples

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

Problem C. Revaccination

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

The Byteland Medical Agency has approved N of different vaccine options. After revaccination with the i-th option, the level of resistance x, determined before the revaccination, changes to $a_i x + b_i$.

For revaccination shall be used exactly two **different** vaccine variants, one for the first step and another for the second step. To make recommendations to physicians on revaccination, it is required to respond to the following types of queries:

Given the initial level of resistance x, determine what the maximum level of resistance can be reached after the completion of the revaccination process (that is after both selected options have been applied).

Input

The first line of the input contains one integer N ($2 \le N \le 10^5$) — the number of vaccines registered with the Byteland Medical Agency.

Each of the following n lines contains two integers a_i and b_i ($-10^6 \le a_i \le 10^6$, $-10^{12} \le b_i \le 10^{12}$. The next line contains one integer Q ($1 \le Q \le 10^5$). Each of the following Q lines contains one integer x_i — the initial level of resistance in i -th query ($-10^6 \le x_i \le 10^6$, all x_i are pairwise distinct).

Output

For each query print one integer — the maximum level of resistance that can be reached after the completion of the revaccination process.

Example

standard input	standard output
4 -2 1 1 1 -2 0 2 1 3 21 -11 -2021	85 47 8087

Problem D. Yandexatellite

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

In order to build maps on Yandex panoramas, special cars equipped with photographic equipment are used that drive along the roads of the country and take pictures of the surrounding landscapes.

But within the framework of the new Yandex.Space project is planned to add the panoramas from deep space to Yandex.Panoramas. It is no longer original to send cars into space, so a special Yandexatellite spacecraft was developed for this. Yandexatellite flies through the asteroid belt and takes pictures using the following algorithm.

1. Initially, the number of images is 0;
2. The neighborhood of the asteroid belt is represented as a subspace of a three-dimensional Cartesian space such that the coordinates x, y, z range from 0 to 10^9.
3. The spaceship flies in an orbit given by the equation $(ax + by + z)^2 + cx + dy + ez + 1 = n$
4. The shooting points are located at the points of the orbit with the following property: all coordinates are non-negative integers, and their product is equal to the given integer L.
5. A picture that was taken at the shooting point with coordinates x, y, and z is transmitted to Earth in x seconds (during these x seconds, a SuperTransmitter is turned on).

Your task is to calculate the total operating time of a SuperTransmitter during the mapping of the asteroid belt.

Input

The first line of the input contains 5 integers — the parameters of the orbit a, b, c, d, e ($1 \le a, b, c, d, e \le 30$).

The next line contains the values n and L ($1 \le n \le 10^9$, $0 \le L \le 10^{18}$).

It is guaranteed that the minimum distance between any two consecutive shooting points in any of the projections does not exceed 200.

Output

Print one integer — the summary time of the SuperTransmitter to work.

Examples

standard input	standard output
3 1 10 8 1 1000 0	1
3 1 10 8 1 1000000000 40019525280	1440

Note

In the second sample, one of the shooting points is — $(1440, 1059, 26243)$.

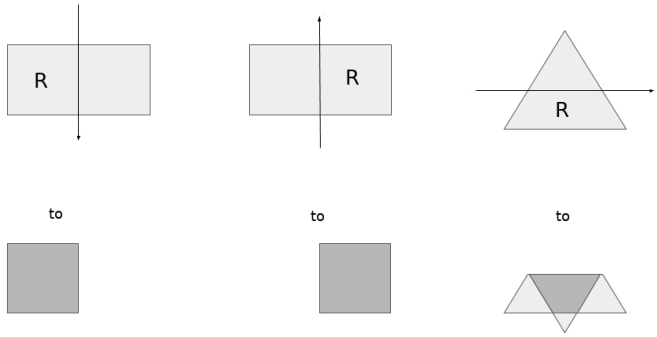
Problem E. A Folding Task

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Bytica is a famous artist. Initially, her art is a convex polygon with N vertices cut from paper.

Bytica did the following action L times:

- Choose a vector v from (x_b, y_b), intersecting the current art along a segment of nonzero length.
- Fold the existing structure along the intersection line to the right side of the direction of the vector v.



In the final version of art, each point corresponds from 1 to 2^L layers paper. For each of the values $1, 2, \ldots 2^L$ output one real number — the total area of the part with exactly i paper layers are overlapping.

Input

The first line of the input contains two integers N and L ($3 \le N \le 30$, $1 \le L \le 5$).

Each of the following N points contains two integers x_i and y_i ($-1000 \le x_i, y_i \le 1000$); those points are given in counterclockwise order.

Each of the following L lines contains four integers x_b, y_b and x_e, y_e, respectively ($-1000 \le x_b, y_b, x_e, y_e \le 1000$) — coordinates start and the end of the corresponding vector.

It is guaranteed that the polygon is convex, that no two vertices of the polygon coincide, no three vertices of the polygon are collinear, that no vector has a beginning and an end at the same point, and that the vector always intersects the current art along a segment of nonzero length.

Output

Print 2^L lines, the i-th of these lines must contain one real number — the area of the part of the structure on which exactly i paper layers overlap, with an absolute or relative error no worse than 10^{-4} ...

Examples

standard input	standard output
4 1 -1 -1 1 -1 1 1 -1 1 -1 0 0 1	3.000000000000 0.500000000000
3 1 0 1 6 1 3 5 0 3 5 3	6.000000000000 3.000000000000
4 2 29 34 69 21 80 82 38 54 49 33 -102 -58 64 59 -602 -25	389.775459017107 492.497392672239 70.243251879471 0.000000000000

Problem F. Bonsai Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Vsevolod is a retired competitive programmer. Now he is working for Researching United Charity of Dendrary Elements (RUCoDE). As a famous member of the Charity, he got a bonsai tree as a birthday gift.

The bonsai is the rooted binary tree with the following property: each vertex of the tree either is the leaf or has exactly two children. The root has an index 1 and all other vertices are enumerated by integers between 2 and N. In each leaf initially the number 1 was written. Consider the value of the vertex as the greatest common divisor of the integers written at the leaves that belong to the subtree with the root in that vertex.

Vsevolod's work is to log the events that happen with the tree. There are two types of events:

1. Multiple the integer from the vertex v by x; you may assume that v is a leaf;
2. Print the value of the vertex v modulo $10^9 + 7$.

Help Vsevolod to handle the events.

Input

The first line of the input contains two integers N and Q ($1 \le N, Q \le 3 \cdot 10^5$) — the number of vertices in the tree and the number of the events, respectively.

The second line contains $N - 1$ integer p_i. i-th of those integers denotes the vertex that is the parent of the vertex $i + 1$ ($1 \le p_i \le N$).

Each of the following M lines describes one event. If the line starts with the integer 1, it means that it is the event of the first type, and then two integers v and x follow ($1 \le v \le N$, $1 \le x \le 10^6$). If the line starts with integer 2, this is the event of the second type, and one integer v follows ($1 \le v \le N$).

Output

For each event of type 2, print the answer modulo $10^9 + 7$.

Example

standard input	standard output
3 10 1 1 1 2 13 1 3 44340 1 3 22553 2 1 2 2 2 3 1 2 2 2 1 2 2 2 3	1 13 13 2 26 13

Problem G. Pac-Man Speedrun

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

The 1C games studios plan to release the unique game "The Siberian Pac-Man" developed for Siemens SL45 or other Rarely Used Communication Devices (RUCoDe). The Siberian Pac-Man is a modification of Pac-Man with the following rules.

The Pac-man starts at the upper-left cell of the $r \times c$ grid. Each cell contains exactly one dot. The first time when Pac-Man visits the cell, the dot is erased. The goal of the game is to erase all the points and finish at the cell at a-th row and b-th column.

Fedor works for 1C games. Also, he is a famous speedrunner, so he plans to speedrun the Siberian Pac-Man on Twitch. To prepare for the speedrun, he asked you to describe the Pac-Man's route as **shortest possible** string '`U`', '`L`', '`D`' and '`R`' (describing the moves up, left, down and right respectively).

The rows are numbered upside down, the columns — from left to right.

Input

The first line of the input contains one integer t ($1 \le t \le 600$) – the number of the test cases.

Each of the following t lines describes one test case and contains four integers r, c, a and b ($2 \le r, c \le 5000$, $1 \le a \le r$, $1 \le b \le c$, $a \cdot b \ne 1$) — number of rows and columns and the row and the column of the target cell, respectively.

You may assume that sum of $r \cdot c$ for all test cases does not exceed $3 \cdot 10^6$.

Output

For each test case print at the separate line one string, composed of characters '`L`', '`R`', '`D`', '`U`' — one of possible shortest paths of Pac-Man to erase all the dots and finish at the target cell.

If there is more than one shortest path, print any of them.

Examples

standard input	standard output
2 3 3 2 2 2 2 1 2	RRDDLLUR DRU
1 2 3 1 3	DRRULR

Problem H. Most Skilled

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

There are N students in group 2021 of the Robotic Universal College of Deep Engineering (RUCoDE), each has a unique id from 1 to N.

On the first day in college, each student thinks that she is most skilled.

Then M lessons happen. Each lesson is encoded by three integers (p, x, y), where $1 \le p, x, y \le N$.

At the start of the lesson, the teacher asks the student with id p about the most skilled student in the group. Let the answer be q. If $p = q$, then the student is assigned with the new project. Otherwise, the student q is asked and so on until the project is assigned or the teacher realized that the same student is asked twice — then the teacher assigns the project to the whole group.

Then, after the lesson, the student x changes the opinion and considers student y as the most skilled one.

For each integer, i between 1 and M find the student with the new project assigned in i-th day. If the project was assigned to the whole group, output -1 instead.

Input

First line of the input contains two integers N and M ($1 \le N, M \le 3 \cdot 10^5$). The following M lines describe the lessons; each of those lines contains three integers p, x and y ($1 \le p, x, y \le N$).

Output

Print M lines. i-th of those lines shall contain two integers — id of the student with the project assigned at i-th day, and the number of questions asked by the teacher before the assignment, if the project was assigned to some student, or single integer -1, if the project was assigned to the whole group.

Example

standard input	standard output
3 4 1 2 3 2 3 2 2 2 1 3 1 2	1 1 3 2 -1 1 3

Problem I. Nice Colorings — 3

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

This is an interactive problem.

Little Ildar got t sheets of paper 100×100 cells and two pencils: black and red. Ildar colored each cell on sheets by one of two colors — black or red and asks his friend Kirill to play the following guessing game. Totally there are t rounds of the game, each round is played on the new sheet.

Kirill may ask Ildar about the color of the cell at the intersection of the named row and the named column. Kirill can ask no more than 15 questions, then he shall choose four cells such as:

- Their centers form the rectangle with sides parallel to the coordinate axis.
- All four cells are painted in the same color.

Kirill asks you to write the program that helps him to win.

Interaction Protocol

The first line of the input contains one integer t ($1 \le t \le 2000$) – the number of the games played by Ildar and Kirill. All those games are **independent** and each game is played on a sheet different than previous ones.

The jury program is playing for Ildar, you are playing for Kirill and asking the questions. Note that the jury program is **adaptive**, i.e. the coloring may be finally chosen after your questions (but the color of the cells that the jury program is told to you will never change).

To ask the jury program about the color of the cell, use the query in the format '`?`' r c, where r and c denote the row and the column of the cell.

The jury program outputs the line with the string "`B`", if the cell is black, or "`R`" if the cell is red. You may ask no more than 15 queries.

When you think that you are ready to name the four cells of the same color, print '`!`' x_1 y_1 x_2 y_2 to tell that the cells (x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2) are colored in the same color ($1 \le x_1, y_1, x_2, y_2 \le 100$, $x_1 \ne x_2$, $y_1 \ne y_2$).

The jury program will answer string "`OK`" to you if your guess is correct, and the next game begins immediately.

If your guess is wrong, if you asked too many queries or used the invalid row or column in the query or answer, the jury program will answer "`FAIL`". After you receive that message, your program shall immediately quit (otherwise you may have the ML, IL, or TL instead of WA).

Example

standard input	standard output
1 R B R R R OK	? 1 1 ? 1 2 ? 1 3 ? 3 1 ? 3 3 ! 1 1 2 2

Note

For the correct work, any request of final answer shall be terminated by line feed character, and after each such output do not forget to clear the output buffer, that is:

- In Pascal: **flush (output)**;
- In C / C ++: **fflush (stdout)** or **cout.flush ()**;
- In Java: **System.out.flush ()**;
- In Python: **sys.stdout.flush ()** from library **sys**;
- In C #: **Console.Out.Flush ()**;

Problem J. Vasily's Ladder

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Old Vasily is an elder follower of the Retro-Urbanistic Cult of Design (RUCoDE). The followers of that constructive cult obey to redesign at least something in their homes or gardens each day.

There are n sticks in the old Vasily's fence. Sticks are placed in a row and enumerated from left to right by the sequential integers between 1 and n, respectively. Each stick has the integer height between 1 and n; two sticks **can** have the same height.

Vasily wants to choose a continuous block of sticks from l-th to r-th inclusively, remove it from the fence and build the ladder.

Vasily may build the ladder if the chosen set of sticks contains exactly one stick of the length 1, exactly one stick of length 2, and so on, i.e. if all sticks in the set have distinct height and the maximum height of the stick in the set is equal to the number of the sticks in the set.

The height of the ladder is defined as the height of the highest stick in the set. Help old Vasily to choose l and r so the ladder will have maximal possible height.

Input

The first line of the input contains one integer n ($1 \le n \le 3 * 10^5$) — the number of the sticks in the fence.

The second line contains n space-separate integers a_i, where a_i is the height of the i-th stick in the fence ($1 \le a_i \le n$).

Output

Print one integer — maximum height of the ladder that may be built from a continuous block of sticks in the fence.

Example

standard input	standard output
19 9 5 2 1 3 4 7 2 5 6 5 8 1 3 7 2 4 10 1	8

Note

If we take sticks 2 to 6, it allows building the ladder of height 5. Sticks 4 to 10 give the set that can form the ladder of height 7, but the highest ladder can be formed from sticks 10 to 17, so the answer is 8.

Problem K. Sandwiches

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Alexander is the last year student of the Regional University of Cooking Delicacies (RUCoDe).

Now Alexander is working on the recipes for sandwiches.

In the beginning, there are K empty sandwiches (pieces of bread) on the table, placed in line. Then Alexander does N operations, each of them has one of the following three types:

1. Alexander takes a slice of cheese of type x from the fridge and puts it at the top of i-th sandwich;
2. Alexander takes the top slice of cheese from i-th sandwich and returns it to the fridge;
3. Alexander thinks that sandwiches i and j differ too much, so he starts to move slices of cheese from the top of the sandwich with the greater number of slices of cheese to top of the sandwich with the lesser number of slices of cheese one by one till the difference between these numbers will be at most 1.

Your task is to calculate the number of distinct sandwiches that were on the table when Alexander was processing those operations. Note that the empty sandwich is not counted, but the sandwiches that appear while performing the operations of type 3 shall be taken into account. Two sandwiches differ if they have different numbers of slices of cheese on them or if at for some i the i-th layers of cheese have different types.

Input

The first line contains two integers K and N ($K \le 20$, $N \le 3 \cdot 10^5$) – the number of empty sandwiches on the table when the process is started and the number of actions, respectively.

Each of the following N lines describes one operation.

The line starts with the operation type — the integer 1, 2 or 3. For the operation of type 1, two integers i and x follow ($1 \le i \le K$, $1 \le x \le 10^9$) — the number of the sandwich and the type of cheese placed on top, respectively. For the operation of type 2, one integer i ($1 \le i \le K$) is written — the number of sandwich to remove the cheese from. You may assume that the given sandwich is non-empty. For the operation of type 3, two integers i and j are given ($1 \le i, j \le K$, $i \ne j$) — numbers of sandwiches to be "equalized".

Output

Print one integer — the answer to the problem.

Example

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

Problem L. Tea With Milk

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Tea without milk is a waste of money

The vending machine has N containers. Each container is filled with a mixture of tea and milk in different proportions, that is, t_i milliliters of tea and m_i milliliters of milk are mixed in the i-th container.

The vending machine can mix drinks from different containers in any proportion (of course, no more than what was originally poured into the container).

That is, if you pour the a_i-th part of the i-th capacity ($0 \le a_i \le 1$), then it turns out a mixture consisting of $\sum\limits_{i=1}^{N} a_i \cdot t_i$ milliliters of tea and $\sum\limits_{i=1}^{N} a_i \cdot m_i$ milliliters of milk.

The customer wants to order $x + y$ milliliters of drink, in the ratio of x milliliters of tea to y milliliters of milk. If the automaton can fulfill this order, then the pair of numbers (x, y) is called feasible.

Given a set of values for t_i and m_i, find the number of feasible pairs, where both x and y the are **positive integers**.

Since the number can be very large, print it modulo $10^9 + 7$.

Input

The first line of the input contains one integer N ($1 \le N \le 10^5$). Each of the following N lines contains two integers t_i and m_i — the amount of tea and milk in the mixture from the i-th container ($1 \le t_i, m_i \le 10^9$).

Output

Print one integer — the number of the feasible pairs modulo $10^9 + 7$.

Example

standard input	standard output
4 1 2 2 4 11 21 20 21	254