Problem A. Krypton

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

In the future several years, you will surely remember that you once participated in a special China Collegiate Programming Contest. Due to COVID-19, all parties had paid a lot of effort to eventually hold a rare online programming contest. As a problem writer, I would also like to express my sincere gratitude to the organizing committee, all the contestants and others who have worked hard in this process. I also sincerely wish you good results in this special contest and a special and sweat memory in the future.

Maybe several years later, you may recall that ...

Once there was a mobile game, where there was a virtual currency called coupon that could only be purchased by RMB. Kelo found that relying on his liver alone cannot make him stronger, so he decided to purchase some coupons. When he opened the recharge page, he found that this game had seven recharge columns, and was holding an activity called "First Recharge Reward". If it was the first time a recharge column was chosen, some additional coupons would be given to the player as a reward. A player might receive several rewards if he chose several different recharge columns. A column could be chosen for arbitrary times, but there would be no additional reward except for the first time it was chosen. Here is a table describing the price, amount of coupons in normal case and additional first recharge rewards of each column.

Price (RMB yuan)	Normal amount (coupons)	First recharge reward (coupons)
1	10	8
6	60	18
28	280	28
88	880	58
198	1980	128
328	3280	198
648	6480	388

Kelo had recently earned n yuan by writing problems for China Collegiate Programming Contest. He decided to recharge all these n yuan to the game, and hoped to get as many coupons as possible with these hard-earned money.

Input

The only line contains an only integer n ($1 \le n \le 2000$).

Output

Print the maximum amount of coupons Kelo might get.

Examples

standard input	standard output
100	1084
198	2108

Problem B. The Tortoise and the Hare

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	256 megabytes

Once there was a hare, who was addicted to sleeping. The hare lived in a town, where there were also n tortoises living in a row on a street, numbered from 1 to n.

The hare could run much faster than the tortoises, so he often challenged the tortoises m-meter run. During such a race, when the hare was just before the finish line, the i-th tortoises could only run a_i meters.

On a race day, the hare would pick an interval, maybe $[l, r]$. The tortoises numbered between l and r (inclusive), together with the hare, would compete in an m-meter race. Just before the the finish line, the hare would sleep for a while, as he led the race so much.

The hare had trained an ability during his long-term sleepiness, which was to curse k tortoises every second when sleeping. The ability was available as soon as he fell asleep. Any cursed tortoise would be trapped in place for a second when cursed. One tortoise might be cursed multiple times. As the hare had no ability to think during sleep, he simply chose the k tortoises closest to the finish line when applying his cursing ability.

After the hare fell asleep, due to the anger caused by the leader sleeping and the companion being trapped, all tortoises not trapped began to run as fast as 1 meter per second.

In order to win the game, if the hare could not trap all the tortoises that were only one meter away from the finish line at some moment, he must wake up and cross the finish line immediately.

The hare wondered how many seconds he could sleep during each race, and asked you for help.

However, the problem was not that easy. On every day when there was no race, the m-meter running speed of one of the turtles will increase due to diligent exercise (or maybe decrease due to lack of exercise). Specifically, after that day, the tortoise numbered u could run v meters before the hare was just before the finish line. Note that his running speed might be changed again in the future.

Now can you still tell the hare exactly how many seconds he could sleep in each race?

Input

The first line contains three integers n, m and d ($2 \le n \le 10^5$, $1 \le m \le 10^9$, $1 \le d \le 10^5$), where d represents the number of days to be concerned.

The second line contains n integers $a_1, \ldots, a_n$ ($1 \le a_i < m$), representing the initial running speed of the tortoises.

Each of the next d lines starts with an integer r ($1 \le r \le 2$), describing one day.

- $r = 1$ indicates a racing day. Three integers l, r and k ($1 \le l < r \le n$, $1 \le k \le r - l$) follow on the same line.
- $r = 2$ indicates a day without racing. Two integers u and v ($1 \le u \le n$, $1 \le v < m$) follow on the same line.

Output

For each racing day, print on a line the maximum seconds the hare could sleep in the race.

Example

standard input	standard output
5 10 6 6 1 4 2 3 1 1 5 3 1 3 5 1 2 2 9 1 1 5 2 1 1 5 4 1 3 5 1	14 9 7 21 9

Note

On the fourth day in the example, when the hare fell asleep, the tortoises each had completed 6, 9, 4, 2 and 3 meters. The tortoises numbered 1 and 2 would be cursed, so one second later, the tortoises would have completed 6, 9, 5, 3 and 4 meters. Repeating this procedure, one may find that all tortoises would have completed 9 meters after 7 seconds, just one meter before the finish line. As $5 > 2 = k$, the hare had to immediately wake up and run across the finish line.

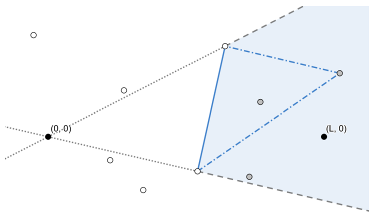
Problem C. Quantum Geometry

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

Once there was a spiritual guy, who wanted to meet his best friend as soon as possible. Both he and his friend were on an Euclidean plane. He started his trip from $(0, 0)$, and his friend was located at $(L, 0)$.

If everything had been normal, he would reach his destination by walking L meters, straight from $(0, 0)$ to $(L, 0)$. However, the region around was influenced by a mysterious force, which might be blamed on the n magic towers around.

Every midnight, three towers were chosen by the mysterious force. High walls would appear on the boundary of the triangle formed by them as vertices, and would stay there until the next midnight. The walls were high enough to block sight, causing the spiritual guy unable to learn what was behind the wall. In other words, if the spiritual guy could only see one wall, any tower blocked by the wall, from his perspective, might be chosen as the remaining third tower. The spiritual guy travelled so fast that he could arrive where his friend were within one day. Therefore, you may assume the 3 towers chosen remained unchanged during his trip. While travelling, he could gradually learn about the unknown area from his new perspective. Fortunately, he had so good a memory and a sense of direction that he knew where the n towers and his destination were even if they were blocked by the wall, and could remember any situation as long as he was once able to see it. The spiritual guy was a pessimist, and would choose the shortest strategy in the worst case.



Example: The spiritual guy knew where the three grey towers and his destination were, but could not determine which one of the three grey towers was chosen when he set off.

If the spiritual guy could see all three chosen towers when departing, choosing a shortest path was no difficulty to him, so he came to you to verify his strategy where he could only see two chosen towers when he set off. He had designed several situations, and you are expected to tell him how much distance he needed to travel in the worst case, so that he could check his own answer.

Input

The first line contains three integers n, L and q ($3 \le n \le 3 \times 10^3, 2 \le L \le 10^8$, $1 \le q \le 3 \times 10^5$), where q represents the number of situations the spiritual guy had designed.

Each of the next n lines contains two integers x and y ($|x| \le 10^8, |y| \le 10^8$), indicating that there was a tower at (x, y).

Each of the next q lines contains two integers u and v ($1 \le u < v \le n$), indicating the situation that only the u-th and v-th towers in the above input could be seen when the spiritual guy set off. It is guaranteed that there was at least one tower blocked by the wall connecting these two towers.

It is also guaranteed that neither the starting point nor the destination might be surrounded by the walls no matter how the mysterious force chose towers. In addition, you may assume that no line would pass at least three of the towers, the starting point and the destination.

Output

For each situation, print on a line the minimum distance needed to be travelled in the worst case.

Your answer is considered correct if its absolute or relative error does not exceed 10^{-9}.

Formally, let your answer be a, and the jury's answer be b. Your answer is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-9}$.

Example

standard input	standard output
5 5 2 1 2 2 2 4 1 1 -2 3 -2 1 4 1 3	6.841619252964 5.000000000000

Note

For the first situation in the example, the spiritual guy would go straight to $(1, 2)$ to have a look of what was behind the first wall. Only at the moment he arrived at $(1, 2)$ could he get some new information, and at that moment the worst case was that $(2, 2)$ was chosen to be the third tower. Another strategy was to go to $(1, -2)$ after departing, the worst case was that $(3, -2)$ was chosen, resulting a longer distance. You may prove that all other strategies also had a worse result.

For the second situation in the example, $(2, 2)$ was the only choice of the third tower. The spiritual guy could walk straight from $(0, 0)$ to $(5, 0)$.

Problem D. Meaningless Sequence

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Once there was a mathematician, who was obsessed with meaningless number sequences. Here is one of them.

$$a_n = \begin{cases} 1, & n = 0 \\ c \cdot \max\limits_{0 \le i < n} a_{n \,\&\, i}, & \text{otherwise} \end{cases},$$

where & denotes the bitwise AND operation.

As a mathematician, he could easily tell what a_n was for any n, but he wanted to test you. You are required to tell him

$$\left(\sum_{i=0}^{n} a_i \right) \bmod (10^9 + 7)$$

to convince him that you have a deep understanding of this (although meaningless) sequence.

Input

The only line contains two integers n and c ($0 \le n < 2^{3000}, 0 \le c \le 10^9$).

Specially, n is given in **binary presentation**. c is given in decimal presentation normally.

Output

Print what you are expected to tell the mathematician normally in decimal presentation.

Example

standard input	standard output
1000 3	67

Problem E. Defense of Valor League

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	256 megabytes

Once there was a highly anticipated e-sports championship, the 2020 *League of Legends* World Championship, where DWG defeated SN 3 to 1 in the finals.

Even though Oscar is a player of the game *Defense of Valor League* (DoVL) — a recently developed MOBA game, he could also felt the excitement of the finals, especially when the player *SN Bin* took the extraordinary PENTA KILL using the champion *Fiora*.

While watching the final games, he noticed that two teams always picked the red side when they were able to choose to act as the blue team or the red team. One possible explanation for such phenomenon may be that the red team holds the last pick, or *Counter Pick*, that may help team choose one good hero countering the opponent's lineup. He fell into deep thought: in the *ban pick phase* of the game, how much advantage or disadvantage does the blue team have?

To investigate the problem he decided to do some experiments in DoVL, which is a p-player versus p-player game. To start a game, each of the two teams need to pick p distinct heroes of all n heroes to form a lineup by *ban pick phase*, as described as follows.

- **Ban Phase** Two teams take b turns to ban a hero (to clarify, **each** team has b turns), starting from the blue team. All heroes banned should be **distinct**. However, teams are allowed to decide not to ban any hero in any of their turns (known as an *empty ban*). The banned heroes can not be chosen in the pick phase by **both** teams.
- **Pick Phase** Two teams take turns to pick heroes to form a lineup. The blue team first picks 1 hero. Then starting from the red team, the two teams take turns to pick 2 heroes, until only one hero is missing in one team's lineup, and that team finally picks their last hero. **All** heroes picked should be **distinct** and **not banned**.

To evaluate the lineup quality of picking, we follow three perspectives to calculate a score.

- **Strength** Some heroes are naturally stronger than others in the current game. If one team picked hero i, they will earn c_{ii} points.
- **Combo** Some heroes will be stronger when working with one another. If two different heroes i and j are in the same team, the team will earn c_{ij} points for such a combination. It's guaranteed that $c_{ij} = c_{ji}$, but you should not double count both c_{ij} and c_{ji}.

- **Counter** Some heroes make some specific opponents hard to perform well. If heroes i and j are in different teams, the team picking i will earn $d_{ij}/2$ points, while the team picking j will earn $d_{ji}/2$. It's guaranteed that $d_{ij} = -d_{ji}$.

Formally, suppose the lineup of the blue team is $(x_1, \ldots, x_p)$ and the one of the red team is $(y_1, \ldots, y_p)$. The total advantage (or disadvantage if negative) of the blue team can be evaluated by

$$(\sum_{i=1}^{p} c_{x_i x_i} + \sum_{i=1}^{p} \sum_{j=i+1}^{p} c_{x_i x_j} + \sum_{i=1}^{p}\sum_{j=1}^{p} \frac{1}{2} d_{x_i y_j}) - (\sum_{i=1}^{p} c_{y_i y_i} + \sum_{i=1}^{p} \sum_{j=i+1}^{p} c_{y_i y_j} + \sum_{i=1}^{p}\sum_{j=1}^{p} \frac{1}{2} d_{y_i x_j}).$$

Given all the information above, can you help Oscar to calculate the advantage (or maybe disadvantage) the blue team will take when the two teams act optimally? It's not an easy task, so you may claim that you outperform the coaches of the world champions if you manage to solve this problem!

Input

Each test contains multiple test cases. The first line contains the number of test cases T $(1 \le T \le 10)$. Description of the test cases follows.

For each test case, the first line contains three integers b, p and n ($p \ge 1$, $b \ge 0$, $b+p \le 5$, $2(b+p) \le n \le 12$).

The i-th of the next n lines contains n integers $c_{i1}, \ldots, c_{in}$ $(0 \le c_{ij} \le 1000)$. It is guaranteed that $c_{ij} = c_{ji}$.

The i-th of the next n lines contains n integers $d_{i1}, \ldots, d_{in}$ $(-1000 \le d_{ij} \le 1000)$. It is guaranteed that $d_{ij} = -d_{ji}$.

It is also guaranteed that, for test cases, where $T \ge 5$ the value of c_{ij} $(i \le j)$ and d_{ij} $(i < j)$ are generated independently uniformly randomly.

Output

For each test case, print on a line an integer representing the advantage (or disadvantage if negative) of the blue team.

Example

standard input	standard output
4 0 2 4 5 0 0 0 0 2 9 0 0 9 2 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 4 4 0 0 0 0 3 0 0 0 0 2 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 3 0 0 0 0 0 0 0 0 0 0 -1 1 1 0 -1 -1 1 0 1 1 4 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 1 0 1 0 -1 0 -1 1 0	-4 1 -1 1

Note

Explanation for the four examples:

1. Even hero 1 is a strong choice, but there's an unstoppable combo of 2 and 3. So the blue team must pick one of these two heroes to minimize disadvantage. In the red team's turn hero 1 and the remaining one of $\{2, 3\}$ will be picked. The disadvantage of the first team is $(2 + 1) - (5 + 2) = -4$.
2. It can be proven that no matter which hero the blue team bans, the red team can always make the strongest two unbanned heroes differ by 1 in strength.
3. Do you remember the game *Rock, Paper, Scissors*? The one who can see the other's choice can always win.
4. The blue team will ban any of $\{2, 3, 4\}$. Otherwise, the red team will act so that only hero 1 is banned (maybe using an *empty ban*), leading to a *Rock, Paper, Scissors* game which puts the blue team at a disadvantage.

Problem F. Strange Memory

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Once there was a rooted tree. The tree contained n nodes, which were numbered $1, \ldots, n$. The node numbered 1 was the root of the tree. Besides, every node i was assigned a number a_i. Your were surprised to find that there were several pairs of nodes (i, j) satisfying

$$a_i \oplus a_j = a_{\mathrm{lca}(i,j)},$$

where $\oplus$ denotes the bitwise XOR operation, and $\mathrm{lca}(i, j)$ is the lowest common ancestor of i and j, or formally, the lowest (i.e. deepest) node that has both i and j as descendants.

Unfortunately, you cannot remember all such pairs, and only remember the sum of $i \oplus j$ for all different pairs of nodes (i, j) satisfying the above property. Note that (i, j) and (j, i) are considered the same here. In other words, you will only be able to recall

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} [a_i \oplus a_j = a_{\mathrm{lca}(i,j)}](i \oplus j).$$

You are assumed to calculate it now in order to memorize it better in the future.

Input

The first line contains a single integer n ($2 \le n \le 10^5$).

The second line contains n integers, $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^6$).

Each of the next $n - 1$ lines contains 2 integers u and v ($1 \le u, v \le n, u \ne v$), indicating that there is an edge between u and v. It is guaranteed that these edges form a tree.

Output

Print what you will memorize in the future.

Example

standard input	standard output
6 4 2 1 6 6 5 1 2 2 3 1 4 4 5 4 6	18

Problem G. Monkey's Keyboard

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

Once there was a monkey, who was smart enough to press keys on a keyboard. However, he could not understand what he pressed. One day, a computer idiot came to the monkey for help. He needed to type a chapter of *Complete Works of Shakespeare*, denoted as a string s. When the monkey kept typing, he focused on the screen. As long as the chapter he wanted occurred on the screen as a substring, he would stop the monkey immediately and gain his result using `ctrl-c` and `ctrl-v`.

For simplicity, we assume that the keyboard only consisted of 26 lowercase English characters, and the computer idiot didn't care about spaces, punctuation and capitalization, so s only consisted of lowercase English characters. As the monkey actually didn't understand what he was typing, he pressed the keys randomly. The probability of the key β pressed on each type was $p_\beta / \sum_{\gamma=a}^{z} p_\gamma$, and independent of what had already been typed.

The computer idiot waited for the Shakespeare's work he wanted on the screen till hungry and cold, so he asked you how long he had to wait in the words of mathematical expectation. The time was measured by the number of keys pressed by the monkey.

The question seemed easy for you, and after learning the result, the computer idiot realized that he might not be able to get Shakespeare's work while alive. Therefore, he decided to get a substring of s. In order to make a more reasonable decision, he asked you again the same problem on every substring of s.

Since there are too many substrings of s, you only need to output the sum of the expected time on each substring.

Throughout the problem, a substring means a continuous subsequence of another string.

Input

The input consists of three lines.

The first line contains a string s $(1 \le |s| \le 5 \times 10^5)$ consisting of lowercase English characters.

Each of the next 2 lines contains 13 integers, $p_a, p_b, \ldots, p_m$ and $p_n, p_o, \ldots, p_z$ respectively $(1 \le p_a, p_b, \ldots, p_z \le 5 \times 10^5)$.

Output

Output the sum, for t being every substring of s, the expected number of keys pressed until t appeared on the screen, modulo $10^9 + 7$.

Formally, let $M = 10^9 + 7$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \bmod M$. In other words, output such an integer x that $0 \le x < M$ and $x \cdot q \equiv p \pmod{M}$.

Examples

standard input	standard output
abc 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	19006
aabccdaabccdaa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26	394656279

Problem H. Combination Lock

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

Once there was a combination lock, which consisted of m rings with digits from 0 to 9. It was broken so one could never open the lock no matter how he spun it. Alice and Bob wanted to play a game with it. They in turn spun one ring of the lock for one step in either direction, and tried to open the lock (although they knew the lock would never open). If the password on the lock had already been tried, the player would be considered lost. Additionally, they had discussed a set S of n passwords before the game started. The player would also lose if (s)he tried a password among S. Alice would move first.

A 5-digit combination lock

Now you can only recall m, S and the initial password on the lock, and you wonder who was the winner supposing Alice and Bob played optimally.

Input

Each test contains multiple test cases. The first line contains the number of test cases T $(1 \leq T \leq 10)$. Description of the test cases follows.

The first line contains two integers m and n $(1 \leq m \leq 5,\ 0 \leq n < 10^m)$, and a string t $(|t| = m)$ composed of digits from 0 to 9, representing the initial password on the lock.

The i-th of the next n lines contains a string u_i $(|u_i| = m)$ composed of digits from 0 to 9, representing a password in S. It is guaranteed that all u_i are distinct, and not the same as t.

Output

For each test case, print on a line `Alice` or `Bob`, the winner of the game.

Example

standard input	standard output
3	Bob
5 3 30583	Bob
29348	Alice
80064	
09637	
1 2 6	
7	
5	
1 2 9	
1	
8	

Problem I. Kawaii Courier

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Once there was a courier named Kujou Kawaii, whose work was to collect packages from people around the city. The city consisted of n communities, numbered from 1 to n, together with $n-1$ bidirectional roads connecting them. The express distribution center of this city was located at the community numbered k. Of course, any two communities were guaranteed to be connected by these roads. Kujou's electromobile was so small that she could only carry packages from 1 community at the same time. She would always first deliver packages from the community numbered 1, then those from the one numbered 2, etc., and finally the packages from the one numbered n.

Let's calculate Kujou's tiredness index after a whole day of work! We only consider the trip from each community to the express distribution center. Unfortunately, Kujou had no sense of direction. On arriving at a community, she would uniformly randomly choose a road (including the one she came from) to follow, until she finally reached the community where the express distribution center was located. Suppose the trip from the community numbered i to the express center passed d_i roads. Kujou's tiredness was influenced by three factors.

1. Tiredness a_i from beginning of the work. a_i equals i.
2. Physical tiredness b_i during the current delivery. b_i equals d_i.
3. Emotional tiredness c_i. Kujor enjoyed traveling all over the city, so c was initialized to 1 for each delivery, and was multiplied by x where x is a rational number in $(0, 1]$ for each road Kujor passes. In other words, c_i equals x^{d_i}.

For each delivery i, Kujou's final tiredness index would be $m_i = a_i \times b_i \times c_i$, and you need to calculate the expected tiredness index for each delivery.

It can be shown that m_i can be expressed as an irreducible fraction $\frac{p_i}{q_i}$, where p_i and q_i are integers. It is then guaranteed in all test cases that $q_i \not\equiv 0 \pmod{10^9+7}$. In this case, m_i modulo 10^9+7 is well-defined, which is some integer m'_i such that $m'_i q_i \equiv p_i \pmod{10^9+7}$. Furthermore, you only need to output $m'_1 \oplus \cdots \oplus m'_n$ to prevent large output, where $\oplus$ denotes the bitwise XOR operation.

Input

The first line contains four integers n, k, p_x and q_x ($2 \le n \le 10^5, 1 \le k \le n, 1 \le p_x \le q_x \le 10^7$), where x is given in the form p_x/q_x.

Each of the next $n-1$ lines contains two integers u and v ($1 \le u, v \le n$), indicating that there was a road connecting the community numbered u and the one numbered v. It is guaranteed that all communities are connected by these $n-1$ roads.

Output

Print $m'_1 \oplus \cdots \oplus m'_n$.

Examples

standard input	standard output
3 1 1 1 1 2 3 1	1
3 1 1 2 1 2 2 3	695000161
5 3 1 2 1 2 1 3 4 3 5 1	1000005122

Note

In the second example, $m_1 = 0$, $m_2 = \frac{36}{49}$, $m_3 = \frac{48}{49}$.

Problem J. Abstract Painting

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Once there was a painter, expert in drawing circles on a canvas. As a tasteful painter, he found that if he drew circles according to the following restrictions, the resulting canvas would look like an abstract painting.

1. The center of any circle must be a grid point on the x-axis.
2. The x-coordinate of any point on any circle must be within the range $[0, n]$.
3. The radius of any circle must be a positive integer not exceeding 5.
4. Any two circles have at most 1 intersection point.

Specially, an empty canvas was also considered an abstract painting, as it broke none of the above restrictions.

There had already been k circles drawn on the canvas, not breaking the above restrictions. The painter could further draw one or more circles on the canvas to get an abstract painting, or he could draw nothing and claim that it was already an excellent abstract painting. He wondered how many different abstract paintings he could possibly get. As the number may be very large, you only need to output it modulo $10^9 + 7$.

Input

The first line contains two integers n and k ($2 \le n \le 10^3$, $0 \le k \le 5n$).

If $k > 0$, each of the next k lines contains two integers c and r ($1 \le c < n, 1 \le r \le \lfloor \frac{n}{2} \rfloor$), indicating that there was already a circle centering at $(c, 0)$ with radius r on the plane.

It is guaranteed that the drawn k circles satisfy the above restrictions.

Output

Print the number of different abstract paintings the painter might get, modulo $10^9 + 7$.

Example

standard input	standard output
4 1 1 1	4

Problem K. Ragdoll

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

Once there was a lovely ragdoll cat, named Little Zara, who liked trees and math. One day she met the doge Adam. Adam had just planted some trees each consisting of only one node. The nodes were numbered from 1 to n, and the i-th node was assigned a value a_i. Adam liked pairing tree nodes, but he disliked some node pairs. A pair of nodes (i, j) was considered bad if i and j were in the same tree and $\gcd(a_i, a_j) = a_i \oplus a_j$, where $\gcd(x, y)$ denotes the greatest common divisor (GCD) of integers x and y, and $\oplus$ denotes the bitwise XOR operation. Adam wondered how many bad pairs there were in his forest.

Zara was good at solving problems about trees and math, so she could answer Adam's question in a short time. However, Adam was so naughty a dog that he would repeatedly change the forest slightly and ask Zara his question again after the change.

The changes Adam might make are listed here:

1. Adam plants a new tree with only one node numbered x and assigned a value v.
2. Adam uses magic to merge the tree containing the node x and the one containing the node y. If x and y are in the same tree before the operation, the magic fails and has no effect.
3. Adam changes the value of node x to v.

Now you are expected to help Zara answer all questions Adam asked, in order that they could sing and dance together happily.

Input

The first line contains two integers n and m ($1 \le n \le 10^5$, $1 \le m \le 2 \times 10^5$), representing the number of nodes in the original forest and the number of changes Adam would make, respectively.

The next line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 2 \times 10^5$).

Each of the next m lines describes a change Adam made, starting with an integer t ($1 \le t \le 3$) representing the type of the change.

- If $t = 1$, it will be followed by two integers x and v ($n < x \le n + m$, $1 \le v \le 2 \times 10^5$). It is guaranteed that x's are distinct in all changes of the first type.
- If $t = 2$, it will be followed by two integers x and y ($1 \le x, y \le n + m$). It is guaranteed that the node x and y already exist in the forest.
- If $t = 3$, it will be followed by two integers x and v. ($1 \le x \le n + m$, $1 \le v \le 2 \times 10^5$). It is guaranteed that the node x already exists in the forest.

Output

For each change Adam made, print one line with a single integer, representing the number of bad pairs in the forest after the change.

Examples

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

Problem L. Coordinate Paper

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Once there was a contest, where the last problem was about coordinate paper. You were given a piece of coordinate paper, where there were grids of n rows and 10^{100} columns. The grids were initially white, and you might paint some of them black. In that problem, you were required to paint the paper in a special way. Suppose the number of black grids in the i-th row is a_i. Your painting was supposed to satisfy that

1. for any i, $a_i \geq 0$.
2. $\sum_{i=1}^{n} a_i = s$.
3. for any $i \in \{1, \ldots, n-1\}$, either $a_i - a_{i+1} = k$ or $a_{i+1} - a_i = 1$.

Can you still find an answer to that problem?

Input

The only line contains three integers n, k and s ($1 \leq n, k \leq 10^5$, $1 \leq s \leq 10^{18}$).

Output

If there is no solution, print "-1" (without quotes).

Otherwise, print the n integers $a_1, \ldots, a_n$. Any answer satisfying all requirements will be accepted.

In this problem, extra blank characters will be ignored when your answer is judged.

Example

standard input	standard output
3 2 15	6 4 5