Problem A. Rectangular Posters

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	512 megabytes

A rectangular board with sides parallel to the coordinate axes lies on a two-dimensional plane. The bottom-left corner of the board has coordinates $(0, 0)$, and the top-right corner has coordinates (W, H).

There are n rectangular posters, the i-th of which has a width of w_i $(1 \le w_i < W)$ and a height of h_i $(1 \le h_i < H)$. These posters are placed fully inside the board randomly and independently, without being rotated or flipped, and the sides are parallel to the coordinate axes. More specifically, the bottom-left corner of the i-th poster has coordinates (x_i, y_i), and the top-right corner has coordinates $(x_i + w_i, y_i + h_i)$, where each x_i is independently and uniformly randomly chosen from $[0, W - w_i]$, and each y_i is independently and uniformly randomly chosen from $[0, H - h_i]$.

You need to find the expected area covered by the n posters modulo $10^9 + 7$.

Input

The first line contains three integers n $(1 \le n \le 120)$, W, and H $(2 \le W, H \le 10^9)$, indicating the number of rectangular posters, the width, and the height of the rectangular board.

Then n lines follow, the i-th of which contains two integers w_i $(1 \le w_i < W)$ and h_i $(1 \le h_i < H)$, indicating the width and the height of the i-th rectangular poster.

Output

Output a line containing an integer, indicating the expected area covered by the n posters modulo $10^9 + 7$.

It can be proved that the probability is always a rational number. Additionally, under the constraints of this problem, it can also be proved that when that value is represented as an irreducible fraction p/q, we have $q \not\equiv 0 \pmod{10^9 + 7}$. Thus, there is a unique integer r $(0 \le r < 10^9 + 7)$ such that $p \times r \equiv q \pmod{10^9 + 7}$. This r is what we need.

Examples

standard input	standard output
1 2 2 1 1	1
2 2 2 1 1 1 1	555555561

Note

For the first sample case, the expected covered area is 1.

For the second sample case, the expected covered area is 14/9.

Problem B. Binary Substrings 2

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

Given two integers n and m, you need to find a binary string of length n such that the number of different nonempty substrings is exactly m. Here, m is no less than n, the minimum number of different nonempty substrings in a binary string of length n, and is no more than $M_n = \sum_{i=1}^{n} \min\{2^i, n-i+1\}$, the proved maximum number.

However, the problem described above seems too hard, so you just need to find a binary string of length n such that the number of different nonempty substrings has at most 0.2 relative error with m, i.e., is in the range $[0.8 \times m, 1.2 \times m]$, or indicate that no binary string meets this condition.

Input

The first line of the input contains an integer T $(1 \le T \le 10^4)$, indicating the number of test cases. For each test case:

The only line contains two integers n $(1 \le n \le 2 \times 10^5)$ and m $(n \le m \le M_n)$.

It is guaranteed that the sum of n for all test cases does not exceed 2×10^5.

Output

For each test case, output a line containing a binary string of length n that meets the condition, or "`-1`" (without quotes) if no binary string meets the condition.

Examples

standard input	standard output
5 5 5 5 6 5 7 5 8 5 9	00000 00000 -1 00001 00001
5 1 1 2 3 3 5 4 8 5 12	0 01 011 0110 01100

Problem C. Clamped Sequence 2

Input file:	`standard input`
Output file:	`standard output`
Time limit:	8 seconds
Memory limit:	512 megabytes

Given an integer sequence $a_1, a_2, \ldots, a_n$, along with a weight sequence $w_1, w_2, \ldots, w_{n-1}$, you need to answer q queries. Each query gives a positive integer d, and you need to clamp the sequence $a_1, a_2, \ldots, a_n$ to a range $[l, r]$ satisfying $0 \le r - l \le d$ that maximizes $\sum_{i=1}^{n-1} w_i \times |a_i - a_{i+1}|$, where $|x|$ is the absolute value of x.

More specifically, clamping the sequence $a_1, a_2, \ldots, a_n$ to the range $[l, r]$ makes each element

$$a_i := \begin{cases} l, & a_i < l; \\ a_i, & l \le a_i \le r; \\ r, & a_i > r. \end{cases}$$

Both l and r are arbitrary real numbers decided by you under the given constraints. It can be shown that the maximum weighted sum is always an integer.

Input

The first line contains two integers n ($2 \le n \le 1\,000$) and q ($1 \le q \le 10^6$), indicating the length of the given sequence and the number of queries.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$), indicating the given sequence.

The third line contains $n-1$ integers $w_1, w_2, \ldots, w_{n-1}$ ($-10^6 \le w_i \le 10^6$), indicating the weight sequence.

Then q lines follow, each containing an integer d ($1 \le d \le 2 \times 10^9$), indicating the given parameter for this query.

Output

Output q lines, each containing a single integer, indicating the maximum weighted sum under the given parameter d.

Example

standard input	standard output
6 3 3 1 4 1 5 9 2 6 5 3 5 2 3 5	32 46 54

Note

For the first query, the clamping range can be $[1, 3]$, so the sequence will be $[3, 1, 3, 1, 3, 3]$, and the value equals $2 \times |3 - 1| + 6 \times |1 - 3| + 5 \times |3 - 1| + 3 \times |1 - 3| + 5 \times |3 - 3| = 4 + 12 + 10 + 6 + 0 = 32$.

For the second query, the clamping range can be $[1, 4]$, so the sequence will be $[3, 1, 4, 1, 4, 4]$, and the value equals $2 \times |3 - 1| + 6 \times |1 - 4| + 5 \times |4 - 1| + 3 \times |1 - 4| + 5 \times |4 - 4| = 4 + 18 + 15 + 9 + 0 = 46$.

For the third query, the clamping range can be $[1, 6]$, so the sequence will be $[3, 1, 4, 1, 5, 6]$, and the value equals $2 \times |3 - 1| + 6 \times |1 - 4| + 5 \times |4 - 1| + 3 \times |1 - 5| + 5 \times |5 - 6| = 4 + 18 + 15 + 12 + 5 = 54$.

Problem D. LCM Challenge

Input file:	`standard input`
Output file:	`standard output`
Time limit:	8 seconds
Memory limit:	512 megabytes

There are lots of things to do in this contest besides this problem, so let's make it quick.

Given two integers n and p, where p is prime, you need to find the value of $\mathrm{LCM}(1, 2, 3, \ldots, n) \bmod p$, i.e., the least common multiple of $1, 2, 3, \ldots, n$ modulo p.

Input

The only line contains two integers n and p ($1 \le n < p < 10^{10}$), where p is prime.

Output

Output a line containing a single integer, indicating the value of $\mathrm{LCM}(1, 2, 3, \ldots, n) \bmod p$.

Examples

standard input	standard output
10 9999999967	2520
30 9999999967	9089570456
9999999966 9999999967	6047288450

Note

For the first sample case, $\mathrm{LCM}(1, 2, 3, \ldots, 10) = 2^3 \times 3^2 \times 5 \times 7 = 2\,520$.

For the second sample case, $\mathrm{LCM}(1, 2, 3, \ldots, 30) = 2\,329\,089\,562\,800$.

Problem E. Endless Ladders

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

In the ancient Square Kingdom, the resident c ($c = 1, 2, 3, \ldots$) lives on a stone pillar that is c^2 units high from the ground.

For the convenience of visiting each other, the Square King built ladders of different lengths. A ladder of length d enables two residents whose absolute difference between their heights is exactly d to visit each other. Due to the limited budget, a ladder of length d is built if and only if there are two residents such that the absolute difference between their heights is exactly d, and only one ladder of the same length will be built.

The ladders are labeled with $1, 2, 3, \ldots$ starting from the shortest length. One day, the resident a wants to visit the resident b, and you need to find the label of the ladder they should use.

Input

The first line of the input contains an integer T ($1 \le T \le 10^4$), indicating the number of test cases. For each test case:

The only line contains two integers a and b ($1 \le a, b \le 10^9, a \ne b$), indicating that the resident a is going to visit the resident b.

Output

For each test case, output a line containing an integer, indicating the label of the used ladder.

Example

standard input	standard output
2 3 1 2 5	4 14

Note

The lengths of the first 5 labeled ladders are $3, 5, 7, 8, 9$ respectively.

For the first sample case, the length of the used ladder is $3^2 - 1^2 = 8$, which is labeled with 4.

Problem F. Flight Tracker 2

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

Let the Earth be a sphere centered at $(0, 0, 0)$ with a radius r in 3D Euclidean space. There is a flight flying along the shortest path from the departure place to the destination place on the surface of the Earth.

As an aviation enthusiast, you have a receiver that can receive the signal from the flight with a distance no more than d. Note that we calculate the distance between two points by measuring the shortest path on the surface of the Earth, which is NOT the Euclidean distance in 3D Euclidean space.

You need to find the area of the region on the surface of the Earth where you can receive the signal from the flight with the receiver at some time when the flight is flying.

Input

The first line of the input contains an integer T ($1 \le T \le 10^4$), indicating the number of test cases. For each test case:

The first line contains two integers r ($1 \le r \le 100$) and d ($1 \le d \le 1\,000$), indicating the radius of the Earth and the maximum distance on the surface of the Earth for receiving the signal from the flight.

The second line contains three integers u, v, and w ($-100 \le u, v, w \le 100, u^2 + v^2 + w^2 > 0$), indicating that the departure place has coordinates $\left(\frac{ru}{\sqrt{u^2+v^2+w^2}}, \frac{rv}{\sqrt{u^2+v^2+w^2}}, \frac{rw}{\sqrt{u^2+v^2+w^2}}\right)$.

The third line contains three integers x, y, and z ($-100 \le x, y, z \le 100, x^2 + y^2 + z^2 > 0$), indicating that the destination place has coordinates $\left(\frac{rx}{\sqrt{x^2+y^2+z^2}}, \frac{ry}{\sqrt{x^2+y^2+z^2}}, \frac{rz}{\sqrt{x^2+y^2+z^2}}\right)$.

It is guaranteed that the departure place and the destination place cannot coincide with each other and cannot be directly opposite each other on the Earth. Therefore, the shortest path from the departure place to the destination place on the surface of the Earth is uniquely determined.

Output

For each test case, output a line containing a real number, indicating the area of the region on the surface of the Earth where you can receive the signal from the flight with the receiver at some time when the flight is flying.

Your answer is acceptable if its absolute or relative error does not exceed 10^{-4}. Formally speaking, suppose that your output is a and the jury's answer is b, and your output is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-4}$.

Example

standard input	standard output
3 10 10 1 0 0 0 1 0 10 20 1 0 0 0 1 0 10 30 1 0 0 0 1 0	553.192486159509631660 1167.025509055589598928 1256.637061435917295360

Problem G. Symmetry Intervals

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

Given a string S of length n, you need to answer q queries.

For each query, a string T and an integer a ($1 \le a \le n - |T| + 1$) are given, and you need to find the number of intervals $[u, v]$ ($1 \le u \le v \le |T|$) such that $S_{a+x-1} = T_x$ holds for every integer $x \in [u, v]$, which are called symmetry intervals. Note that $|T|$ is the length of T.

Input

The first line contains two integers n and q ($1 \le n, q \le 10^5$), indicating the length of the given string S and the number of queries.

The second line contains the given string S of length n.

The next q lines each contain a string T ($1 \le |T| \le n$) and an integer a ($1 \le a \le n - |T| + 1$), indicating a query.

It is guaranteed that the sum of $|T|$ does not exceed 10^6 and that all input strings only contain lowercase English letters.

Output

For each query, output one line containing one integer, indicating the number of symmetry intervals.

Example

standard input	standard output
10 3 helloworld follow 1 echo 2 nowgold 4	10 2 6

Note

For the first query, the 10 symmetry intervals are $[3, 3]$, $[3, 4]$, $[3, 5]$, $[3, 6]$, $[4, 4]$, $[4, 5]$, $[4, 6]$, $[5, 5]$, $[5, 6]$, and $[6, 6]$.

For the second query, the 2 symmetry intervals are $[1, 1]$ and $[4, 4]$.

For the third query, the 6 symmetry intervals are $[2, 2]$, $[2, 3]$, $[3, 3]$, $[6, 6]$, $[6, 7]$, and $[7, 7]$.

Problem H. Symmetry Intervals 2

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

Given a binary string S of length n, you need to answer q queries, where each query is in one of the following two types:

1. Given two integers l, r $(1 \le l \le r \le n)$, flip the binary bit S_i for each $i \in [l, r]$.
2. Given three integers l, a, b $(1 \le l \le n, 1 \le a, b \le n - l + 1)$, you need to find the number of intervals $[u, v]$ $(1 \le u \le v \le l)$ such that $S_{a+x-1} = S_{b+x-1}$ holds for every integer $x \in [u, v]$, which are called symmetry intervals.

Input

The first line contains two integers n and q $(1 \le n, q \le 10^6)$, indicating the length of the given string S and the number of queries.

The second line contains the given binary string S of length n.

The next q lines each contain a query, which is in one of the following two types:

- $1\ l\ r$ $(1 \le l \le r \le n)$, indicating that for each binary bit S_i $(i \in [l, r])$, flip the binary bit S_i for each $i \in [l, r]$.
- $2\ l\ a\ b$ $(1 \le l \le n, 1 \le a, b \le n - l + 1)$, indicating that you need to find the number of intervals $[u, v]$ $(1 \le u \le v \le l)$ such that $S_{a+x-1} = S_{b+x-1}$ holds for every integer $x \in [u, v]$.

It is guaranteed that the number of type-2 queries does not exceed 2 500.

Output

For each type-2 query, output a line containing an integer, indicating the number of symmetry intervals.

Example

standard input	standard output
10 3 1001001001 2 4 3 5 1 2 6 2 5 2 6	2 7

Note

For the first query, $S_{3..6} = 0100$, $S_{5..8} = 0010$, the 2 symmetry intervals are $[1, 1]$ and $[4, 4]$.

After the second query, S becomes 1110111001.

For the third query, $S_{2..6} = 11011$, $S_{6..10} = 11001$, the 7 symmetry intervals are $[1, 1]$, $[1, 2]$, $[1, 3]$, $[2, 2]$, $[2, 3]$, $[3, 3]$, and $[5, 5]$.

Problem I. Iron Bars Cutting

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

There are n iron bars, where the length of the i-th iron bar is a_i. These n iron bars are welded together in the order of $1, 2, 3, \ldots, n$ to form a very long iron bar for some usage. Welding two adjacent iron bars creates a weld point, resulting in a total of $n - 1$ weld points.

Little Q needs to cut this long iron bar back into n iron bars. Each time, he can choose an iron bar that has at least one weld point and select a weld point to cut the iron bar into two at that weld point, then let the lengths of the resulting two iron bars be l_1 and l_2. The imbalance of this cut is defined as $|l_1 - l_2|$, and the cost of the cut is defined as $\min\{l_1, l_2\} \times \lceil \log_2(l_1 + l_2) \rceil$. Note that $|x|$ is the absolute value of x, and $\lceil \log_2(y) \rceil$ is the smallest integer z such that $2^z \geq y$.

Little Q hopes that the imbalances of the $n - 1$ cuts $b_1, b_2, \ldots, b_{n-1}$ satisfy $b_1 \geq b_2 \geq \cdots \geq b_{n-1}$, and the total cost of these $n - 1$ cuts is minimized. You need to find the minimum total cost for the first cut at the weld point between the i-th and $(i + 1)$-th iron bars, or indicate if it is impossible to cut out n iron bars, for each $i = 1, 2, \ldots, n - 1$.

Input

The first line of the input contains an integer T ($1 \leq T \leq 200$), indicating the number of test cases. For each test case:

The first line contains an integer n ($2 \leq n \leq 420$), indicating the number of iron bars.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^9$), indicating the lengths of the iron bars.

It is guaranteed that the sum of n for all test cases does not exceed 420.

Output

For each test case, output a line containing $n - 1$ integers, the i-th of which indicates the minimum total cost for the first cut at the weld point between the i-th and $(i + 1)$-th iron bars, or -1 if it is impossible to cut out n iron bars.

Example

standard input	standard output
3 3 8 9 7 3 1 5 9 3 3 1 4	68 75 24 -1 -1 -1

Problem J. Museum Construction

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

There is a big museum consisting of n rooms and some bidirectional corridors. There are at most 3 doors in each room, and the corridor outgoing from the room is behind the door. All corridors outgoing from a single room lead to different rooms. The whole museum is connected, i.e., it is possible to walk between any two rooms, possibly passing through other rooms along the way.

You are to help in setting labels on doors that will make the tour through the whole museum much easier. The idea is that if a room u has d_u doors leading through corridors to other rooms, these doors are labeled with numbers $1, 2, \ldots, d_u$, then all visitors will follow a simple procedure. If they are in room u at the very beginning of their tour, they will choose the door labeled with 1 and pass through the corresponding corridor. If they are in room u and they entered it from the corridor through the door labeled with i, they will pick the door labeled with the next number (i.e., $i + 1$ if $i < d_u$ and 1 if $i = d_u$) and pass through the corresponding corridor.

You need to find a labeling to ensure that the visitors will pass through each corridor at least once irrespective of the room they start the tour in, assuming they follow the rules, do not get bored easily, and walk long enough.

Input

The first line contains an integer n ($3 \le n \le 2 \times 10^5$), indicating the number of rooms in the museum.

The next n lines contain a description of all corridors, the u-th of which describes corridors connecting the u-th room with others. It begins with an integer d_u ($1 \le d_u \le 3$), the number of doors in this room. Then d_u integers $v_1, v_2, \ldots, v_{d_u}$ ($1 \le v_i \le n$, $v_i \ne u$, and $v_i \ne v_j$ if $i \ne j$) follow, giving numbers of rooms that those doors lead to.

Note that all corridors are bidirectional, so if there is a door from room u to room v, there is a door from room v to room u as well.

Output

Output n lines, the i-th of which contains the numbers of rooms directly connected by corridors with room i in the order of their assigned labels.

It can be shown that a valid labeling of doors always exists. If there are multiple valid labelings, you may output any.

Examples

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

Problem K. Museum Acceptance

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

There is a big museum consisting of n rooms and some bidirectional corridors. There are at most 3 doors in each room, and the corridor outgoing from the room is behind the door. All corridors outgoing from a single room lead to different rooms. The whole museum is connected, i.e., it is possible to walk between any two rooms, possibly passing through other rooms along the way.

You are to help in setting labels on doors that will make the tour through the whole museum much easier. The idea is that if a room u has d_u doors leading through corridors to other rooms, these doors are labeled with numbers $1, 2, \ldots, d_u$, then all visitors will follow a simple procedure. If they are in room u at the very beginning of their tour, they will choose the door labeled with 1 and pass through the corresponding corridor. If they are in room u and they entered it from the corridor through the door labeled with i, they will pick the door labeled with the next number (i.e., $i + 1$ if $i < d_u$ and 1 if $i = d_u$) and pass through the corresponding corridor.

Now we have already set a labeling, and you need to find the number of different corridors that the visitors will pass through if they start the tour in each room, assuming they follow the rules, do not get bored easily, and walk long enough.

Input

The first line contains an integer n ($3 \le n \le 2 \times 10^5$), indicating the number of rooms in the museum.

The next n lines contain a description of all corridors, the u-th of which describes corridors connecting the u-th room with others. It begins with an integer d_u ($1 \le d_u \le 3$), the number of doors in this room. Then d_u integers $v_1, v_2, \ldots, v_{d_u}$ ($1 \le v_i \le n$, $v_i \ne u$, and $v_i \ne v_j$ if $i \ne j$) follow, giving numbers of rooms that those doors lead to, in the order of their assigned labels.

Note that all corridors are bidirectional, so if there is a door from room u to room v, there is a door from room v to room u as well.

Output

Output n lines, the i-th of which contains the number of different corridors that the visitors will pass through if they start the tour in room i.

Examples

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

Problem L. Numb Numbers

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

There are n numbers $a_1, a_2, \ldots, a_n$ in a group, labeled with $1, 2, \ldots, n$, and they keep competing with each other. A number competes with any other number in the group every day and loses when it is smaller than its competitor. It takes part in $n-1$ competitions in total, and if it loses at least $\lceil \frac{n-1}{2} \rceil$ competitions, it is numb with failure. Note that $\lceil x \rceil$ is the smallest integer y such that $y \geq x$.

So every day, there can always be some numbers that are numb with failure. As a kind mental therapist, you feel obliged to talk to them to cheer them up. So you wonder how many numbers are numb each day, which determines the amount of your work.

The numbers don't remain the same. Each day, exactly one of them practices really hard and enlarges itself. Once the number changes, it won't change until it further enlarges itself. So every day, you may face a different situation.

Input

The first line of the input contains an integer T ($1 \leq T \leq 10^4$), indicating the number of test cases. For each test case:

The first line contains two integers n ($3 \leq n \leq 2 \times 10^5$) and q ($1 \leq q \leq 2 \times 10^5$), indicating the number of numbers in the group and the number of days for updating their values.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^9$), indicating the value of each number.

Then q lines follow, each of which contains two integers p ($1 \leq p \leq n$) and v ($1 \leq v \leq 10^9$), indicating the label of the number that is enlarged and the value by which it is increased.

It is guaranteed that both the sum of n and the sum of q for all test cases do not exceed 5×10^5.

Output

For each test case, output q integers, indicating the number of numb numbers after each update.

Example

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

Note

For the first sample case:

After the first update, the numbers are $1, 3, 3, 4, 5$, where $1, 3, 3$ feel numb.

After the second update, the numbers are $1, 3, 5, 4, 5$, where $1, 3, 4$ feel numb.

After the third update, the numbers are $1, 4, 5, 4, 5$, where $1, 4, 4$ feel numb.