Problem A. All Possible Increasing Subsequences

Time limit 3000 ms
Mem limit 65536 kB

An increasing subsequence from a sequence $\{A_1, A_2 \dots A_n\}$ is defined by $\{A_{i1}, A_{i2} \dots A_{ik}\}$, where the following properties hold:

1. $i_1 < i_2 < i_3 < \dots < i_k$, and
2. $A_{i1} < A_{i2} < A_{i3} < \dots < A_{ik}$.

Now you are given a sequence, you have to find the number of all possible increasing subsequences.

Input

Input starts with an integer **T (≤ 10)**, denoting the number of test cases.

Each case contains an integer **n ($1 \le n \le 10^5$)** denoting the number of elements in the initial sequence. The next line will contain **n** integers separated by spaces, denoting the elements of the sequence. Each of these integers will be fit into a 32 bit signed integer.

Output

For each case of input, print the case number and the number of possible increasing subsequences modulo **1000000007 ($10^9 + 7$)**.

Sample

Input	Output
3 3 1 1 2 5 1 2 1000 1000 1001 3 1 10 11	Case 1: 5 Case 2: 23 Case 3: 7

Note

1. For the first case, the increasing subsequences are `(1)`, `(1, 2)`, `(1)`, `(1, 2)`, `2`.
2. Dataset is huge, use faster I/O methods.

Problem B. Binary Simulation

Time limit 2000 ms
Mem limit 65536 kB

Given a binary number, we are about to do some operations on the number. Two types of operations can be here:

1. `I i j`, inverts all the bits from **i** to **j** (inclusive).
2. `Q i` return whether the $\mathbf{i}^{\mathbf{th}}$ bit is 0 or 1.

The MSB (most significant bit) is the first bit **(i.e. i=1)**. The binary number can contain leading zeroes.

Input

Input starts with an integer **T (≤ 10)**, denoting the number of test cases.

Each case starts with a line containing a binary integer having length **n** $(\mathbf{1 \le n \le 10^5})$. The next line will contain an integer **q (1 ≤ q ≤ 50000)** denoting the number of queries.

Each query will be either in the form `I i j` where **i, j** are integers and **1 ≤ i ≤ j ≤ n**. Or the query will be in the form `Q i` where **i** is an integer and **1 ≤ i ≤ n**.

Output

For each case, print the case number in a single line. Then for each query `Q i` you have to print 1 or 0 depending on the $\mathbf{i}^{\mathbf{th}}$ bit.

Sample

Input	Output
2 0011001100 6 I 1 10 I 2 7 Q 2 Q 1 Q 7 Q 5 1011110111 6 I 1 10 I 2 7 Q 2 Q 1 Q 7 Q 5	Case 1: 0 1 1 0 Case 2: 0 0 0 1

Note

Dataset is huge, use faster I/O methods.

Problem C. Curious Robin Hood

Time limit 2000 ms
Mem limit 131072 kB

Robin Hood likes to loot rich people since he helps the poor people with this money. Instead of keeping all the money together he does another trick. He keeps **n** sacks where he keeps this money. The sacks are numbered from **0** to **n-1**.

Now each time he can he can do one of the three following tasks:

1. Give all the money of the i^{th} sack to the poor, leaving the sack empty.
2. Add new amount (given in input) in the i^{th} sack.
3. Find the total amount of money from i^{th} sack to j^{th} sack.

Since he is not a programmer, he seeks your help.

Input

Input starts with an integer **T (≤ 5)**, denoting the number of test cases.

Each case contains two integers **n (1 ≤ n ≤ 10^5)** and **q (1 ≤ q ≤ 50000)**. The next line contains **n** space separated integers in the range **[0, 1000]**. The i^{th} integer denotes the initial amount of money in the i^{th} sack **(0 ≤ i < n)**.

Each of the next **q** lines contains a task in one of the following form:

- **1 i** - give all the money of the i^{th}.
- **2 i v** - add money **v (1 ≤ v ≤ 1000)** to the i^{th} **(0 ≤ i < n)** sack.
- **3 i j** - find the total amount of money from the i^{th} sack to the j^{th} sack **(0 ≤ i ≤ j < n)**.

Output

For each test case, print the case number first. If the query type is **1**, then print the amount of money given to the poor. If the query type is **3**, print the total amount from i^{th} to j^{th} sack.

Sample

Input	Output
1 5 6 3 2 1 4 5 1 4 2 3 4 3 0 3 1 2 3 0 4 1 1	Case 1: 5 14 1 13 2

Note

Dataset is huge, use faster I/O methods.

Problem D. Yaroslav and Divisors

Time limit	2000 ms
Mem limit	262144 kB
Input file	stdin
Output file	stdout

Yaroslav has an array $p = p_1, p_2, \ldots, p_n$ $(1 \le p_i \le n)$, consisting of n distinct integers. Also, he has m queries:

- Query number i is represented as a pair of integers l_i, r_i $(1 \le l_i \le r_i \le n)$.
- The answer to the query l_i, r_i is the number of pairs of integers q, w $(l_i \le q, w \le r_i)$ such that p_q is the divisor of p_w.

Help Yaroslav, answer all his queries.

Input

The first line contains the integers n and m $(1 \le n, m \le 2 \cdot 10^5)$. The second line contains n distinct integers $p_1, p_2, \ldots, p_n$ $(1 \le p_i \le n)$. The following m lines contain Yaroslav's queries. The i-th line contains integers l_i, r_i $(1 \le l_i \le r_i \le n)$.

Output

Print m integers — the answers to Yaroslav's queries in the order they appear in the input.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Examples

Input	Output
1 1 1 1 1	1

Input	Output
10 9 1 2 3 4 5 6 7 8 9 10 1 10 2 9 3 8 4 7 5 6 2 2 9 10 5 10 4 10	27 14 8 4 2 1 2 7 9

Problem E. New Year Domino

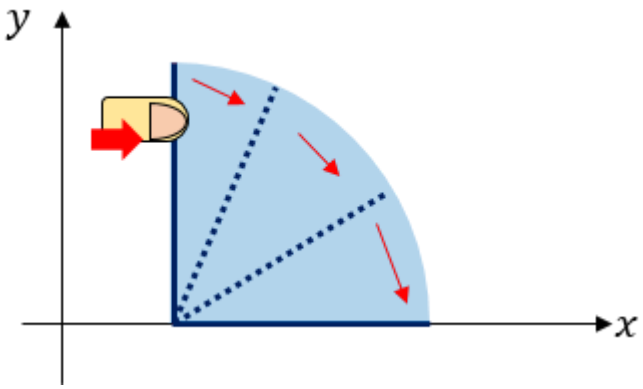
Time limit	2000 ms
Mem limit	262144 kB
Input file	stdin
Output file	stdout

Celebrating the new year, many people post videos of falling dominoes; Here's a list of them: `https://www.youtube.com/results?search_query=New+Years+Dominos`

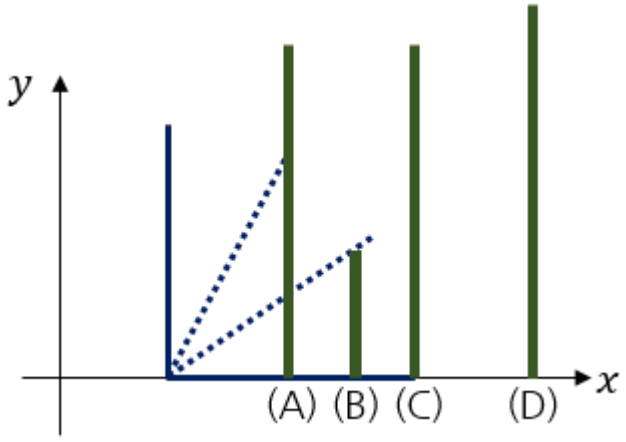
User ainta, who lives in a 2D world, is going to post a video as well.

There are n dominoes on a 2D Cartesian plane. i-th domino ($1 \le i \le n$) can be represented as a line segment which is parallel to the y-axis and whose length is l_i. The lower point of the domino is on the x-axis. Let's denote the x-coordinate of the i-th domino as p_i. Dominoes are placed one after another, so $p_1 < p_2 < \ldots < p_{n-1} < p_n$ holds.

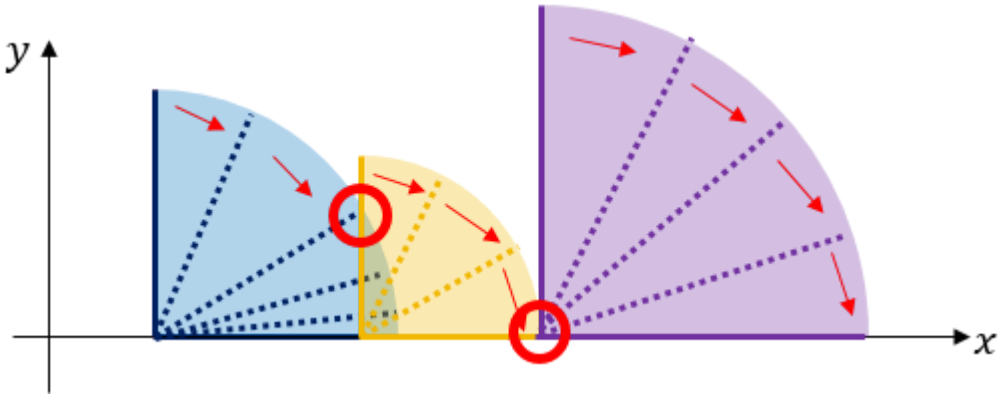
User ainta wants to take a video of falling dominoes. To make dominoes fall, he can push a single domino to the right. Then, the domino will fall down drawing a circle-shaped orbit until the line segment totally overlaps with the x-axis.



Also, if the s-th domino touches the t-th domino while falling down, the t-th domino will also fall down towards the right, following the same procedure above. Domino s touches domino t if and only if the segment representing s and t intersects.



See the picture above. If he pushes the leftmost domino to the right, it falls down, touching dominoes (A), (B) and (C). As a result, dominoes (A), (B), (C) will also fall towards the right. However, domino (D) won't be affected by pushing the leftmost domino, but eventually it will fall because it is touched by domino (C) for the first time.



The picture above is an example of falling dominoes. Each red circle denotes a touch of two dominoes.

User ainta has q plans of posting the video. j-th of them starts with pushing the x_j-th domino, and lasts until the y_j-th domino falls. But sometimes, it could be impossible to achieve such plan, so he has to lengthen some dominoes. It costs one dollar to increase the length of a single domino by 1. User ainta wants to know, for each plan, the minimum cost needed to achieve it. Plans are processed independently, i. e. if domino's length is increased in some plan, it doesn't affect its length in other plans. Set of dominos that will fall except x_j-th domino and y_j-th domino doesn't matter, but the initial push should be on domino x_j.

Input

The first line contains an integer n ($2 \le n \le 2 \times 10^5$)— the number of dominoes.

Next n lines describe the dominoes. The i-th line ($1 \le i \le n$) contains two space-separated integers p_i, l_i ($1 \le p_i, l_i \le 10^9$)— the x-coordinate and the length of the i-th domino. It is guaranteed that $p_1 < p_2 < \ldots < p_{n-1} < p_n$.

The next line contains an integer q ($1 \le q \le 2 \times 10^5$) — the number of plans.

Next q lines describe the plans. The j-th line ($1 \le j \le q$) contains two space-separated integers x_j, y_j ($1 \le x_j < y_j \le n$). It means the j-th plan is, to push the x_j-th domino, and shoot a video until the y_j-th domino falls.

Output

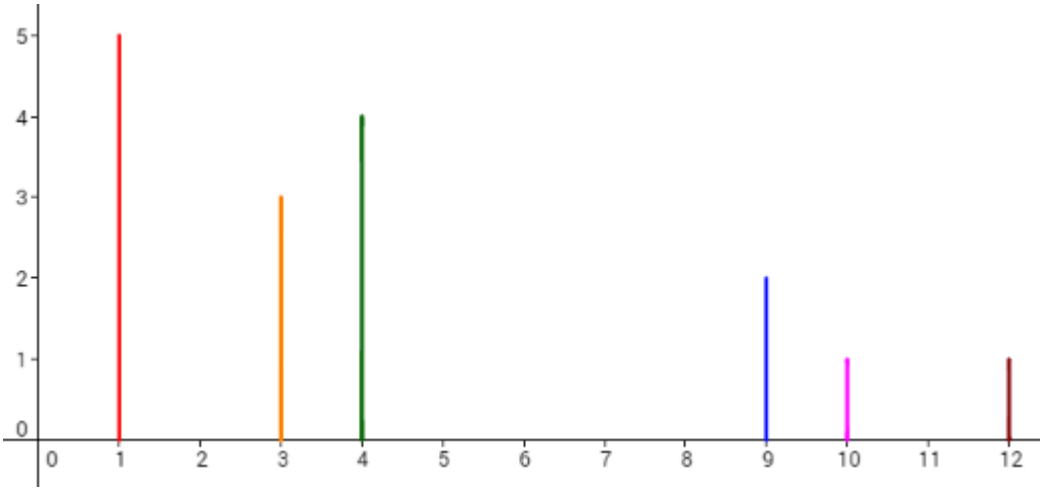
For each plan, print a line containing the minimum cost needed to achieve it. If no cost is needed, print 0.

Examples

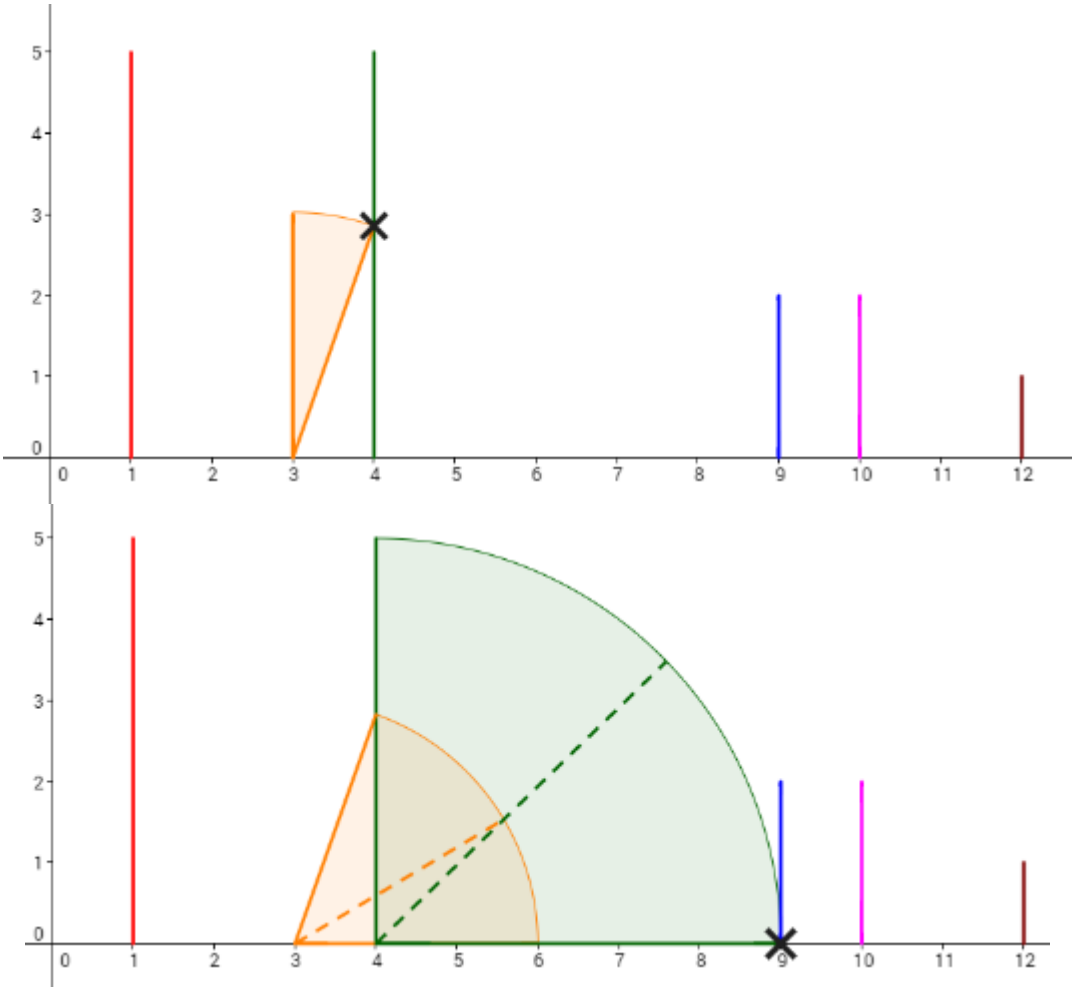
Input	Output
6 1 5 3 3 4 4 9 2 10 1 12 1 4 1 2 2 4 2 5 2 6	0 1 1 2

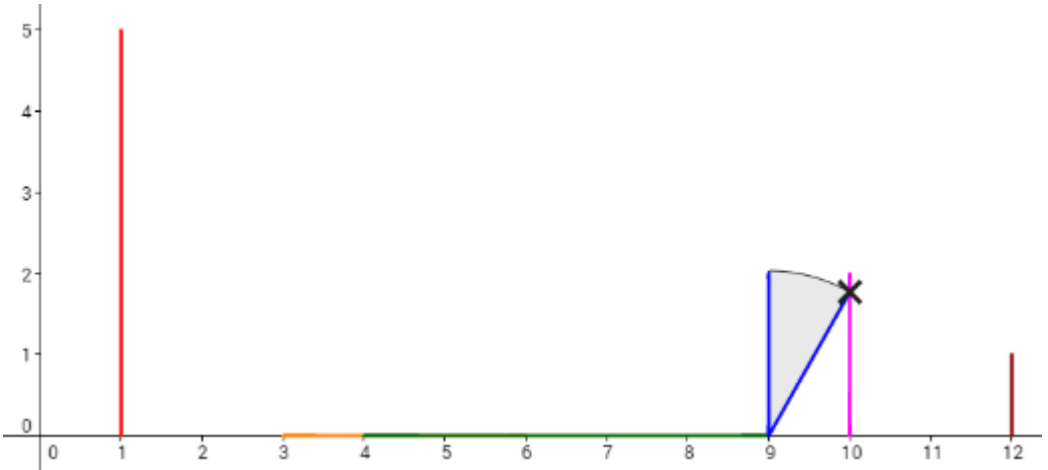
Note

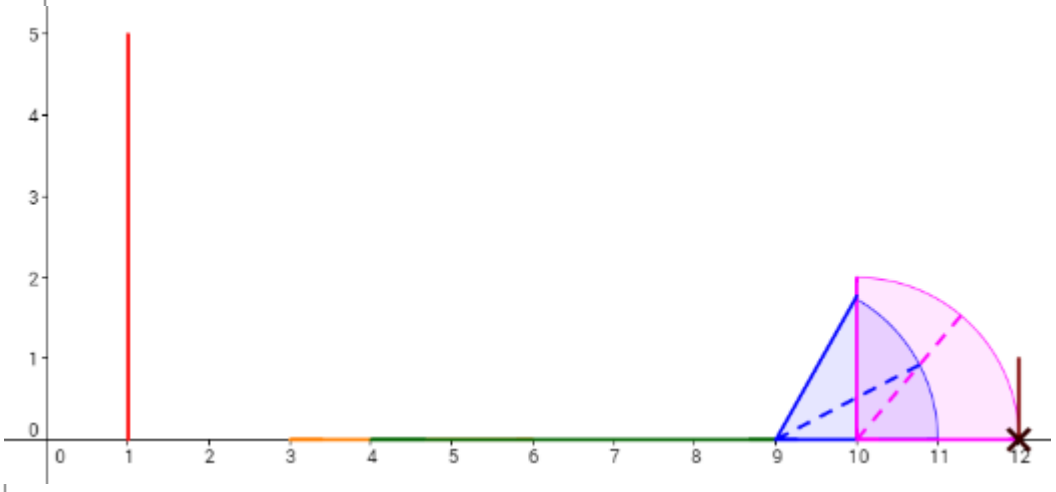
Consider the example. The dominoes are set like the picture below.

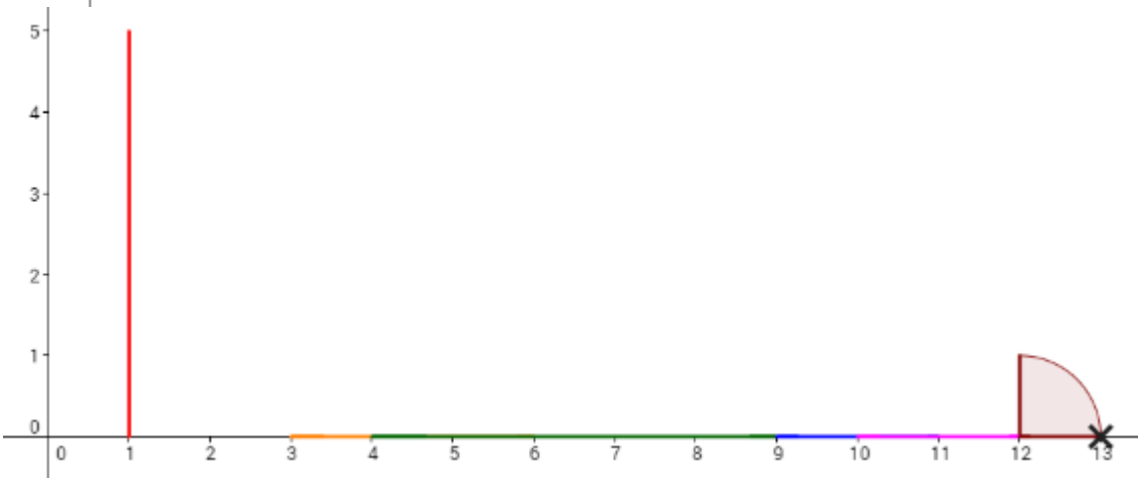


Let's take a look at the 4th plan. To make the 6th domino fall by pushing the 2nd domino, the length of the 3rd domino (whose x-coordinate is 4) should be increased by 1, and the 5th domino (whose x-coordinate is 9) should be increased by 1 (other option is to increase 4th domino instead of 5th also by 1). Then, the dominoes will fall like in the picture below. Each cross denotes a touch between two dominoes.




0 1 2 3 4 5 6 7 8 9 10 11 12
0 1 2 3 4 5


0 1 2 3 4 5 6 7 8 9 10 11 12
0 1 2 3 4 5


0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 1 2 3 4 5

Problem F. Vika and Segments

Time limit 2000 ms
Mem limit 262144 kB

Vika has an infinite sheet of squared paper. Initially all squares are white. She introduced a two-dimensional coordinate system on this sheet and drew n black horizontal and vertical segments parallel to the coordinate axes. All segments have width equal to 1 square, that means every segment occupy some set of neighbouring squares situated in one row or one column.

Your task is to calculate the number of painted cells. If a cell was painted more than once, it should be calculated exactly once.

Input

The first line of the input contains a single integer n ($1 \le n \le 100\,000$) — the number of segments drawn by Vika.

Each of the next n lines contains four integers x_1, y_1, x_2 and y_2 ($-10^9 \le x_1, y_1, x_2, y_2 \le 10^9$) — the coordinates of the endpoints of the segments drawn by Vika. It is guaranteed that all the segments are parallel to coordinate axes. Segments may touch, overlap and even completely coincide.

Output

Print the number of cells painted by Vika. If a cell was painted more than once, it should be calculated exactly once in the answer.

Examples

Input	Output
3 0 1 2 1 1 4 1 2 0 3 2 3	8

Input	Output
4 -2 -1 2 -1 2 1 -2 1 -1 -2 -1 2 1 2 1 -2	16

Note

In the first sample Vika will paint squares $(0, 1)$, $(1, 1)$, $(2, 1)$, $(1, 2)$, $(1, 3)$, $(1, 4)$, $(0, 3)$ and $(2, 3)$.

Problem G. Subsequences

Time limit 1000 ms
Mem limit 262144 kB

For the given sequence with n different elements find the number of increasing subsequences with $k + 1$ elements. It is guaranteed that the answer is not greater than $8 \cdot 10^{18}$.

Input

First line contain two integer values n and k ($1 \le n \le 10^5, 0 \le k \le 10$) — the length of sequence and the number of elements in increasing subsequences.

Next n lines contains one integer a_i ($1 \le a_i \le n$) each — elements of sequence. All values a_i are different.

Output

Print one integer — the answer to the problem.

Examples

Input	Output
5 2 1 2 3 5 4	7

Problem H. Circular RMQ

Time limit	1500 ms
Mem limit	262144 kB
Input file	stdin
Output file	stdout

You are given circular array $a_0, a_1, ..., a_{n-1}$. There are two types of operations with it:

- $inc(lf, rg, v)$ — this operation increases each element on the segment $[lf, rg]$ (inclusively) by v;
- $rmq(lf, rg)$ — this operation returns minimal value on the segment $[lf, rg]$ (inclusively).

Assume segments to be circular, so if $n = 5$ and $lf = 3, rg = 1$, it means the index sequence: $3, 4, 0, 1$.

Write program to process given sequence of operations.

Input

The first line contains integer n ($1 \le n \le 200000$). The next line contains initial state of the array: $a_0, a_1, ..., a_{n-1}$ ($-10^6 \le a_i \le 10^6$), a_i are integer. The third line contains integer m ($0 \le m \le 200000$), m — the number of operartons. Next m lines contain one operation each. If line contains two integer lf, rg ($0 \le lf, rg \le n - 1$) it means rmq operation, it contains three integers lf, rg, v ($0 \le lf, rg \le n - 1; -10^6 \le v \le 10^6$) — inc operation.

Output

For each rmq operation write result for it. Please, do not use `%lld` specificator to read or write 64-bit integers in C++. It is preffered to use `cout` (also you may use `%I64d`).

Examples

Input	Output
4 1 2 3 4 4 3 0 3 0 -1 0 1 2 1	1 0 0

Problem I. Interesting Array

Time limit	1000 ms
Mem limit	262144 kB
Input file	stdin
Output file	stdout

We'll call an array of n non-negative integers $a[1], a[2], \ldots, a[n]$ *interesting*, if it meets m constraints. The i-th of the m constraints consists of three integers l_i, r_i, q_i $(1 \le l_i \le r_i \le n)$ meaning that value $a[l_i] \ \& \ a[l_i + 1] \ \& \ \ldots \ \& \ a[r_i]$ should be equal to q_i.

Your task is to find any *interesting* array of n elements or state that such array doesn't exist.

Expression $x\&y$ means the bitwise AND of numbers x and y. In programming languages C++, Java and Python this operation is represented as "&", in Pascal — as "and".

Input

The first line contains two integers n, m $(1 \le n \le 10^5, 1 \le m \le 10^5)$ — the number of elements in the array and the number of limits.

Each of the next m lines contains three integers l_i, r_i, q_i $(1 \le l_i \le r_i \le n, 0 \le q_i < 2^{30})$ describing the i-th limit.

Output

If the *interesting* array exists, in the first line print "YES" (without the quotes) and in the second line print n integers $a[1], a[2], \ldots, a[n]$ $(0 \le a[i] < 2^{30})$ decribing the *interesting* array. If there are multiple answers, print any of them.

If the *interesting* array doesn't exist, print "NO" (without the quotes) in the single line.

Examples

Input	Output
3 1 1 3 3	YES 3 3 3

Input	Output
3 2 1 3 3 1 3 2	NO

Problem J. Valera and Queries

Time limit	2000 ms
Mem limit	524288 kB
Input file	stdin
Output file	stdout

Valera loves segments. He has recently come up with one interesting problem.

The Ox axis of coordinates has n segments, the i-th segment starts in position l_i and ends in position r_i (we will mark it as $[l_i, r_i]$). Your task is to process m queries, each consists of number cnt_i and a set of cnt_i coordinates of points located on the Ox axis. The answer to the query is the number of segments, such that each of them contains at least one point from the query. Segment $[l, r]$ *contains* point q, if $l \le q \le r$.

Valera found the solution of this problem too difficult. So he asked you to help him. Help Valera.

Input

The first line contains two integers n, m ($1 \le n, m \le 3 \cdot 10^5$) — the number of segments on the axis of coordinates and the number of queries.

Next n lines contain the descriptions of the segments. The i-th line contains two positive integers l_i, r_i ($1 \le l_i \le r_i \le 10^6$) — the borders of the i-th segment.

Next m lines contain the description of the queries, one per line. Each line starts from integer cnt_i ($1 \le cnt_i \le 3 \cdot 10^5$) — the number of points in the i-th query. Then the line contains cnt_i distinct positive integers $p_1, p_2, \ldots, p_{cnt_i}$ ($1 \le p_1 < p_2 < \ldots < p_{cnt_i} \le 10^6$) — the coordinates of points in the i-th query.

It is guaranteed that the total number of points in all queries doesn't exceed $3 \cdot 10^5$.

Output

Print m non-negative integers, where the i-th number is the response to the i-th query.

Examples

Input	Output
3 3 1 3 4 5 6 7 3 1 4 7 2 4 5 1 8	3 1 0

Problem K. XOR on Segment

Time limit	4000 ms
Mem limit	262144 kB
Input file	stdin
Output file	stdout

You've got an array a, consisting of n integers $a_1, a_2, ..., a_n$. You are allowed to perform two operations on this array:

1. Calculate the sum of current array elements on the segment $[l, r]$, that is, count value $a_l + a_{l+1} + ... + a_r$.
2. Apply the xor operation with a given number x to each array element on the segment $[l, r]$, that is, execute $a_l = a_l \oplus x, a_{l+1} = a_{l+1} \oplus x, \ldots, a_r = a_r \oplus x$. This operation changes exactly r - $l + 1$ array elements.

Expression $x \oplus y$ means applying bitwise xor operation to numbers x and y. The given operation exists in all modern programming languages, for example in language *C++* and *Java* it is marked as "^", in *Pascal* — as "`xor`".

You've got a list of m operations of the indicated type. Your task is to perform all given operations, for each sum query you should print the result you get.

Input

The first line contains integer n ($1 \le n \le 10^5$) — the size of the array. The second line contains space-separated integers $a_1, a_2, ..., a_n$ ($0 \le a_i \le 10^6$) — the original array.

The third line contains integer m ($1 \le m \le 5 \cdot 10^4$) — the number of operations with the array. The i-th of the following m lines first contains an integer t_i ($1 \le t_i \le 2$) — the type of the i-th query. If $t_i = 1$, then this is the query of the sum, if $t_i = 2$, then this is the query to change array elements. If the i-th operation is of type 1, then next follow two integers l_i, r_i ($1 \le l_i \le r_i \le n$). If the i-th operation is of type 2, then next follow three integers l_i, r_i, x_i ($1 \le l_i \le r_i \le n$, $1 \le x_i \le 10^6$). The numbers on the lines are separated by single spaces.

Output

For each query of type 1 print in a single line the sum of numbers on the given segment. Print the answers to the queries in the order in which the queries go in the input.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams, or the `%I64d` specifier.

Examples

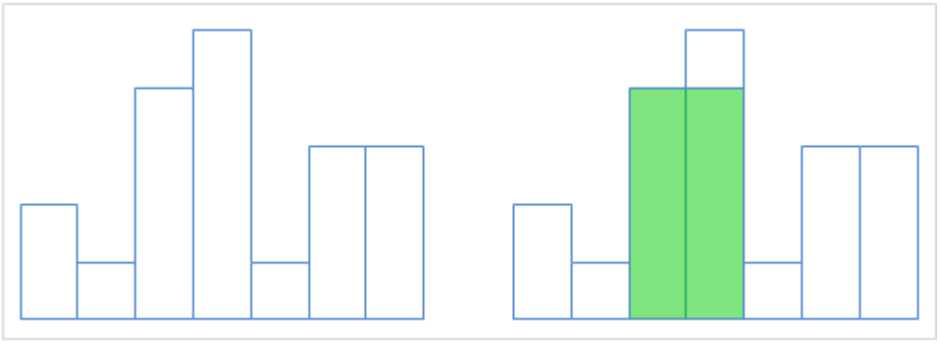
Input	Output
5 4 10 3 13 7 8 1 2 4 2 1 3 3 1 2 4 1 3 3 2 2 5 5 1 1 5 2 1 2 10 1 2 3	26 22 0 34 11

Input	Output
6 4 7 4 0 7 3 5 2 2 3 8 1 1 5 2 3 5 1 2 4 5 6 1 2 3	38 28

Problem L. Histogram

Time limit 2000 ms
Mem limit 65536 kB

A histogram is a polygon composed of a sequence of rectangles aligned at a common base line. The rectangles have equal widths but may have different heights. For example, the figure shows the histogram that consists of rectangles with the heights `{2, 1, 4, 5, 1, 3, 3}` measured in units where the width of the rectangles is 1.



Usually, histograms are used to represent discrete distributions, e.g., the frequencies of characters in texts. Note that the order of the rectangles, i.e., their heights, is important. Calculate the area of the largest rectangle in a histogram that is aligned at the common base line, too. The figure on the right shows the largest aligned rectangle for the depicted histogram.

Input

Input starts with an integer **T (≤ 20)**, denoting the number of test cases.

Each case contains a line with an integer **N (1 ≤ N ≤ 30000)** denoting the number of rectangles. The next line contains **N** space separated positive integers **(≤ 30000)** denoting the heights.

Output

For each case, print the case number and the largest rectangle that can be made.

Sample

Input	Output
2 7 2 1 4 5 1 3 3 5 4 4 3 2 4	Case 1: 8 Case 2: 10

Note

Dataset is huge; use faster I/O methods.

Problem M. Count the Multiples of 3

Time limit 3000 ms
Mem limit 65536 kB

You have an array with **n** elements which is indexed from **0** to **n - 1**. Initially all elements are zero. Now you have to deal with two types of operations:

1. Increase the numbers between indices **i** and **j** (inclusive) by **1**. This is represented by the command **'0 i j'**.
2. Answer how many numbers between indices **i** and **j** (inclusive) are divisible by **3**. This is represented by the command **'1 i j'**.

Input

Input starts with an integer **T (≤ 5)**, denoting the number of test cases.

Each case starts with a line containing two integers **n (1 ≤ n ≤ 10^5)** and **q (1 ≤ q ≤ 50000)** denoting the number of queries. Each query will be either in the form **'0 i j'** or **'1 i j'** where **i**, **j** are integers and **0 ≤ i ≤ j < n**.

Output

For each case, print the case number first. Then for each query in the form **'1 i j'**, print the desired result.

Sample

Input	Output
1 10 9 0 0 9 0 3 7 0 1 4 1 1 7 0 2 2 1 2 4 1 8 8 0 5 8 1 6 9	Case 1: 2 3 0 2

Note

Dataset is huge, use faster I/O methods.

Problem N. Points in Segments (II)

Time limit 2000 ms
Mem limit 65536 kB

Given **n** segments (1 dimensional) and **q** points, for each point you have to find the number of segments which contain that point. A point p_i will lie on a segment **A B** if $A \le p_i \le B$.

For example, if the segments are **(6 12), (8 8), (10 12), (8 11), (0 12)** and the point is **11**, then it is contained by **4** segments.

Input

Input starts with an integer **T (≤ 5)**, denoting the number of test cases.

Each case starts with a line containing two integers **n (1 ≤ n ≤ 50000)** and **q (1 ≤ q ≤ 50000)**.

Each of the next **n** lines contains two integers A_k B_k $(0 \le A_k \le B_k \le 10^8)$ denoting a segment.

Each of the next **q** lines contains an integer denoting a point. Each value will be in $[0, 10^8]$.

Output

For each case, print the case number in a single line. Then for each point, print the number of segments that contain that point.

Sample

Input	Output
1 5 4 6 12 8 8 10 12 8 11 0 12 11 12 2 20	Case 1: 4 3 1 0

Note

Dataset is huge, use faster I/O methods.

Problem O. Counting Primes

Time limit	1000 ms
Mem limit	1572864 kB
Code length Limit	50000 B
OS	Linux

Tortoise and Achilles are playing the Counting the Primes game. Achilles will give Tortoise some numbers, and some intervals, and then Tortoise needs to count the primes on those intervals. It is an easy game, but Tortoise is doing the counting slowly. Achilles is pissed off, so he has given you the task as you are a good programmer. For a twist, he has changed the game a little bit, that is he will give some intervals for counting the prime as well as he will give some intervals to change the numbers in that interval.

You are given an array of **n** elements. After that you will be given **M** commands. They are:

- **0 x y v** - you have to change all numbers in the range of **x** to **y** (inclusive) to **v**, where **x** and **y** are two indexes of the array.
- **1 x y** - output a line containing a single integer which is the number of primes between **x** and **y** (inclusive).

The array is indexed from **1** to **n**.

Input

Input starts with an integer **T (≤ 10)**, denoting the number of test cases.

Each case contains two integers **n (1 ≤ n ≤ 10^4)** and **q (1 ≤ q ≤ 2×10^4)**. Then next line, you will be given **N** integers. After that each of the next **q** lines will contain a task in one of the following form:

- **0 x y v (1 ≤ x ≤ y ≤ n, 2 ≤ v ≤ 10^6)**
- **1 x y (1 ≤ x ≤ y ≤ n)**

And the numbers will be in range of **[2, 10^6]**.

Output

For each case, print the case number first. Then for each query **'1 x y'**, print the number of primes between **x** and **y [inclusively]**.

Example

Input	Output
1 5 3 78 2 13 12 3 1 1 2 0 4 4 5 1 1 5	Case 1: 1 4

Note:

- Use Faster IO like scanf, printf
- A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself. The first prime numbers are 2, 3, 5, 7, 11 ...

Problem P. Ada and Behives

Time limit	2000 ms
Mem limit	1572864 kB
Code length Limit	50000 B
OS	Linux

Ada the Ladybug is currently doing her thesis. It is almost complete with one tiny exception - there are some graphs and statistics missing. The topic of thesis is "**Behavior of Bee Hives**". She examines population of bees and their growth in given areas.

Ada has all data she needs - but parsing it manually might take many long months. She decided to ask you for help. Basically - given population of individual bee hives you will have to answer the number of bees in given area. There are two kinds of queries:

- Query of kind **1** gives you coordinates of hive and number of new-born bees.
- Query of kind **2** gives you description of rectangle. You will be asked to find the number of bees living in it.

Input

The first line contains three integer numbers $1 \le N, M \le 2000, 1 \le Q \le 10^5$, the size of examined area (number of rows and number of columns), and number of queries.

The next N lines contains M integer numbers $1 \le A_{i,j} \le 10^4$, the sizes of hives.

Afterward, Q lines (of two types) follow

First kind $1\ I\ J\ P$, $1 \le I \le N$, $1 \le J \le M$, $1 \le P \le 10^4$, the position of hive and the number of new-born bees.

Second kind $2\ I_1\ J_1\ I_2\ J_2$, $1 \le I_1 \le I_2 \le N$, $1 \le J_1 \le J_2 \le M$, the boundaries of rectangular area for which you want to know the number of bees (more specifically the lower-left and upper right corners).

Output

For each query of second kind, output the number of bees.

Example

Input	Output
6 5 8 1 2 3 4 5 1 2 3 4 5 6 6 6 6 6 5 4 3 2 1 5 4 3 2 1 6 6 6 6 6 2 1 1 6 5 2 1 1 2 4 2 4 2 5 4 1 5 4 4 1 1 1 665 2 1 1 6 5 2 1 1 2 4 2 4 2 5 4	120 20 18 789 685 22

Problem Q. Ada and Species

Time limit	6500 ms
Mem limit	1572864 kB
Code length Limit	50000 B
OS	Linux

As you might know, Ada the Ladybug is a farmer. She needs to choose some species of vegetables. Each vegetable has four important attributes. We say that a vegetable is worse than another vegetable, if all of its four attributes are greater.

She wants to eliminate the list of vegetables, so only vegetables, which are not worse than any other vegetables remains.

Input

The first line contains integer $\mathbf{1 \le N \le 2 \times 10^5}$

Each of the next **N** lines contains four integers $\mathbf{1 \le X, Y, Z, W \le N}$. It is guaranteed, that all **X** attributes are distinct for all vegetables. The same is true for **Y**, **Z** and **W** (so in fact, there are four permutations of numbers from **1** to **N**).

Output

Print the number of vegetables, which are not worse than any other vegetable.

Example Input 1

```
3
1 1 1 1
2 2 2 2
3 3 3 3
```

Example Output 1

```
1
```

Example Input 2

```
10
8 9 9 2
3 7 2 4
5 5 10 10
```

```
9 3 5 9
4 6 8 6
2 8 1 7
1 2 6 1
7 4 7 5
6 1 3 8
10 10 4 3
```

Example Output 2

```
5
```

Problem R. Fountains

Time limit 2000 ms
Mem limit 262144 kB

Arkady plays Gardenscapes a lot. Arkady wants to build two new fountains. There are n available fountains, for each fountain its beauty and cost are known. There are two types of money in the game: coins and diamonds, so each fountain cost can be either in coins or diamonds. No money changes between the types are allowed.

Help Arkady to find two fountains with maximum total beauty so that he can buy both at the same time.

Input

The first line contains three integers n, c and d ($2 \le n \le 100\,000$, $0 \le c, d \le 100\,000$) — the number of fountains, the number of coins and diamonds Arkady has.

The next n lines describe fountains. Each of these lines contain two integers b_i and p_i ($1 \le b_i, p_i \le 100\,000$) — the beauty and the cost of the i-th fountain, and then a letter "C" or "D", describing in which type of money is the cost of fountain i: in coins or in diamonds, respectively.

Output

Print the maximum total beauty of exactly two fountains Arkady can build. If he can't build two fountains, print 0.

Examples

Input	Output
3 7 6 10 8 C 4 3 C 5 6 D	9

Input	Output
2 4 5 2 5 C 2 1 D	0

Input	Output
3 10 10 5 5 C 5 5 C 10 11 D	10

Note

In the first example Arkady should build the second fountain with beauty 4, which costs 3 coins. The first fountain he can't build because he don't have enough coins. Also Arkady should build the third fountain with beauty 5 which costs 6 diamonds. Thus the total beauty of built fountains is 9.

In the second example there are two fountains, but Arkady can't build both of them, because he needs 5 coins for the first fountain, and Arkady has only 4 coins.

Problem S. How to Handle the Fans

Time limit	1000 ms
Mem limit	1572864 kB
Code length Limit	50000 B
OS	Linux

Trey Parker and Matt Stone, the creators of “South Park” are having some problems handling their fans. The number of fans is so huge that can’t even count them properly. So they hired “N” employees for counting the fans. All the “N” employees had their own separate offices and they were located in a straight line with positions numbered as 1, 2, 3 ... up to N. Fans can come to the office of any employee at any time and tell them how they feel about the show and if they are lucky enough, they may get to meet Trey Parker and Matt Stone.

All the employees keep on updating Trey and Matt about the number of fans currently in their offices, so at each moment, they will have a list of “N” positions and the number of fans in each of these positions. Trey and Matt suddenly start taking a walk from office at position “A” to position “B” to meet their fans, but before they start walking they want to know the sum of all the fans in the offices from position “A” to “B”. But counting them one by one is taking a lot of time, so now they hired you, an awesome software engineer to do this task. Your task is to find the sum of all the fans present in the offices between positions “A” to “B” ("A" and "B" inclusive). Let’s see if you could do it fast enough.

Input

The first line of Input contains two integers “N” and “Q”. “N” is the number of employees hired by Trey and Matt. “Q” is the number of queries to be followed.

Each of the next “Q” lines contain a query. A query can be of two types:

“add P F” – this means that “F” number of fans came to the office at Position “P”

“find A B” – this means that Trey and Matt wants to know the sum of fans present at offices at positions “A” to “B”

Output

For each query of the type “find A B”, output the sum of fans present at offices at positions “A” to “B” in a different line.

Constraints

$1 <= N <= 10^6$

$1 <= Q <= 10^5$

$1 <= A < B <= N$

$1 <= P <= N$

$1 <= F <= 10^4$

Example

Input	Output
10 10 find 1 5 add 5 8 add 6 2 find 4 5 find 4 6 add 2 4 find 2 6 add 6 7 find 1 6 find 7 10	0 8 10 14 21 0

Problem T. Its a Murder!

Time limit	1259 ms
Mem limit	1572864 kB
Code length Limit	50000 B
OS	Linux

Once detective Saikat was solving a murder case. While going to the crime scene he took the stairs and saw that a number is written on every stair. He found it suspicious and decides to remember all the numbers that he has seen until now. While remembering the numbers he found that he can find some pattern in those numbers. So he decides that for each number on the stairs he will note down the sum of all the numbers previously seen on the stairs which are smaller than the present number. Calculate the sum of all the numbers written on his notes diary.

Input

First line gives T, number of test cases. 2T lines follow.

First line gives you the number of stairs N.

Next line gives you N numbers written on the stairs.

Output

For each test case output one line giving the final sum for each test case.

Constraints

$T \leq 10$

$1 \leq N \leq 10^5$

All numbers will be between 0 and 10^6.

Example

Input	Output
1 5 1 5 3 6 4	15

Problem U. Ada and Greenflies

Time limit	2000 ms
Mem limit	1572864 kB
Code length Limit	50000 B
OS	Linux

As you might already know, Ada the Ladybug is a farmer. Beside of growing vegetable she also breeds greenflies. There are many kinds of greenflies and it is not trivial to breed them since they behave very strangely. Each kind of greenfly is known under a number. As you have multiple greenflies in a single coop, they produce as much juice as the greatest common divisor of their numbers.

Ada owns a row of greenflies (of several kinds) and she can split them by a fence, so that always a continuous segment of greenflies will be in a same coop

Ada wants to find out the sum of production of greenflies over all possible continuous segments (of size at least one), can you help her?

Input

The first line contains an integer $1 \le N \le 3*10^5$, the number of greenflies in row.

The second line contains N integers $0 \le a_i \le 10^6$

Output

Print a single line: the sum of production of all continuous segments.

Example

Input	Output
5 2 2 3 6 2	29

Example Input 2

```
7
2 4 6 12 18 6 4
```

Example Output 2

```
118
```

Example Input 3

```
5
2 4 8 16 32
```

Example Output 3

```
114
```

Example Input 4

```
4
12 20 20 5
```

Example Output 4

```
96
```

Problem V. Domino Principle

Time limit	2000 ms
Mem limit	262144 kB
Input file	stdin
Output file	stdout

Vasya is interested in arranging dominoes. He is fed up with common dominoes and he uses the dominoes of different heights. He put n dominoes on the table along one axis, going from left to right. Every domino stands perpendicular to that axis so that the axis passes through the center of its base. The i-th domino has the coordinate x_i and the height h_i. Now Vasya wants to learn for every domino, how many dominoes will fall if he pushes it to the right. Help him do that.

Consider that a domino falls if it is touched strictly above the base. In other words, the fall of the domino with the initial coordinate x and height h leads to the fall of all dominoes on the segment $[x + 1, x + h - 1]$.

Input

The first line contains integer n ($1 \le n \le 10^5$) which is the number of dominoes. Then follow n lines containing two integers x_i and h_i ($-10^8 \le x_i \le 10^8, 2 \le h_i \le 10^8$) each, which are the coordinate and height of every domino. No two dominoes stand on one point.

Output

Print n space-separated numbers z_i — the number of dominoes that will fall if Vasya pushes

the i-th domino to the right (including the domino itself).

Examples

Input	Output
4 16 5 20 5 10 10 18 2	3 1 4 1

Input	Output
4 0 10 1 5 9 10 15 10	4 1 2 1

Problem W. Nikita and stack

Time limit 2000 ms
Mem limit 262144 kB

Nikita has a stack. A stack in this problem is a data structure that supports two operations. Operation `push(x)` puts an integer x on the top of the stack, and operation `pop()` deletes the top integer from the stack, i. e. the last added. If the stack is empty, then the operation `pop()` does nothing.

Nikita made m operations with the stack but forgot them. Now Nikita wants to remember them. He remembers them one by one, on the i-th step he remembers an operation he made p_i-th. In other words, he remembers the operations in order of some permutation $p_1, p_2, \ldots, p_m$. After each step Nikita wants to know what is the integer on the top of the stack after performing the operations he have already remembered, in the corresponding order. Help him!

Input

The first line contains the integer m ($1 \le m \le 10^5$) — the number of operations Nikita made.

The next m lines contain the operations Nikita remembers. The i-th line starts with two integers p_i and t_i ($1 \le p_i \le m$, $t_i = 0$ or $t_i = 1$) — the index of operation he remembers on the step i, and the type of the operation. t_i equals 0, if the operation is `pop()`, and 1, is the operation is `push(x)`. If the operation is `push(x)`, the line also contains the integer x_i ($1 \le x_i \le 10^6$) — the integer added to the stack.

It is guaranteed that each integer from 1 to m is present exactly once among integers p_i.

Output

Print m integers. The integer i should equal the number on the top of the stack after performing all the operations Nikita remembered on the steps from 1 to i. If the stack is empty after performing all these operations, print `-1`.

Examples

Input	Output
2 2 1 2 1 0	2 2

Input	Output
3 1 1 2 2 1 3 3 0	2 3 2

Input	Output
5 5 0 4 0 3 1 1 2 1 1 1 1 2	-1 -1 -1 -1 2

Note

In the first example, after Nikita remembers the operation on the first step, the operation `push(2)` is the only operation, so the answer is 2. After he remembers the operation `pop()` which was done before `push(2)`, answer stays the same.

In the second example, the operations are `push(2)`, `push(3)` and `pop()`. Nikita remembers them in the order they were performed.

In the third example Nikita remembers the operations in the reversed order.

Problem X. Maximum Sum

Time limit	1000 ms
Mem limit	1572864 kB
Code length Limit	50000 B
OS	Linux

You are given a sequence A[1], A[2] ... A[N] ($0 \le A[i] \le 10^8$, $2 \le N \le 10^5$). There are two types of operations and they are defined as follows:

Update:

This will be indicated in the input by a 'U' followed by space and then two integers i and x.

U i x, $1 \le i \le N$, and x, $0 \le x \le 10^8$.

This operation sets the value of A[i] to x.

Query:

This will be indicated in the input by a 'Q' followed by a single space and then two integers i and j.

Q x y, $1 \le x < y \le N$.

You must find i and j such that $x \le i, j \le y$ and $i \ne j$, such that the sum A[i]+A[j] is maximized. Print the sum A[i]+A[j].

Input

The first line of input consists of an integer **N** representing the length of the sequence. Next line consists of N space separated integers A[i]. Next line contains an integer **Q**, $Q \le 10^5$, representing the number of operations. Next Q lines contain the operations.

Output

Output the maximum sum mentioned above, in a separate line, for each Query.

Example

Input	Output
5 1 2 3 4 5 6 Q 2 4 Q 2 5 U 1 6 Q 1 5 U 1 7 Q 1 5	7 9 11 12

Warning: large Input/Output data, be careful with certain languages

Problem Y. Welfare State

Time limit 2000 ms
Mem limit 262144 kB

There is a country with n citizens. The i-th of them initially has a_i money. The government strictly controls the wealth of its citizens. Whenever a citizen makes a purchase or earns some money, they must send a receipt to the social services mentioning the amount of money they currently have.

Sometimes the government makes payouts to the poor: all citizens who have strictly less money than x are paid accordingly so that after the payout they have exactly x money. In this case the citizens don't send a receipt.

You know the initial wealth of every citizen and the log of all events: receipts and payouts. Restore the amount of money each citizen has after all events.

Input

The first line contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the numer of citizens.

The next line contains n integers a_1, a_2, ..., a_n ($0 \le a_i \le 10^9$) — the initial balances of citizens.

The next line contains a single integer q ($1 \le q \le 2 \cdot 10^5$) — the number of events.

Each of the next q lines contains a single event. The events are given in chronological order.

Each event is described as either `1 p x` ($1 \le p \le n$, $0 \le x \le 10^9$), or `2 x` ($0 \le x \le 10^9$). In the first case we have a receipt that the balance of the p-th person becomes equal to x. In the second case we have a payoff with parameter x.

Output

Print n integers — the balances of all citizens after all events.

Examples

Input	Output
4 1 2 3 4 3 2 3 1 2 2 2 1	3 2 3 4

Input	Output
5 3 50 2 1 10 3 1 2 0 2 8 1 3 20	8 8 20 8 10

Note

In the first example the balances change as follows: $1\ 2\ 3\ 4 \to 3\ 3\ 3\ 4 \to 3\ 2\ 3\ 4 \to 3\ 2\ 3\ 4$

In the second example the balances change as follows: $3\ 50\ 2\ 1\ 10 \to 3\ 0\ 2\ 1\ 10 \to 8\ 8\ 8\ 8\ 10 \to 8\ 8\ 20\ 8\ 10$

acm International Collegiate Programming Contest 2017 | icpc.foundation

ACM International Collegiate Programming Contest 2017

Latin American Regional Contests

November 10th-11th, 2017

Contest Session

This problem set contains 13 problems; pages are numbered from 1 to 26.

This problem set is used in simultaneous contests hosted in the following countries:

Argentina, Bolivia, Brasil, Chile, Colombia, Costa Rica, Cuba, El Salvador
México, Panamá, Perú, República Dominicana and Venezuela

General information

Unless otherwise stated, the following conditions hold for all problems.

Program name

1. Your solution must be called *codename*`.c`, *codename*`.cpp`, *codename*`.java`, *codename*`.py2` or *codename*`.py3`, where *codename* is the capital letter which identifies the problem.

Input

1. The input must be read from standard input.
2. The input consists of a single test case, which is described using a number of lines that depends on the problem. No extra data appear in the input.
3. When a line of data contains several values, they are separated by *single* spaces. No other spaces appear in the input. There are no empty lines.
4. The English alphabet is used. There are no letters with tildes, accents, diaereses or other diacritical marks (ñ, Ã, é, Ì, ô, Ü, ç, etcetera).
5. Every line, including the last one, has the usual end-of-line mark.

Output

1. The output must be written to standard output.
2. The result of the test case must appear in the output using a number of lines that depends on the problem. No extra data should appear in the output.
3. When a line of results contains several values, they must be separated by *single* spaces. No other spaces should appear in the output. There should be no empty lines.
4. The English alphabet must be used. There should be no letters with tildes, accents, diaereses or other diacritical marks (ñ, Ã, é, Ì, ô, Ü, ç, etcetera).
5. Every line, including the last one, must have the usual end-of-line mark.
6. To output real numbers, round them to the closest rational with the required number of digits after the decimal point. Test case is such that there are no ties when rounding as specified.

Development team

The following persons helped to develop the problem set by creating and improving statements, solutions, test cases and input and output checkers:

Alejandro Strejilevich de Loma, Argentina

Guilherme Albuquerque Pinto, Brasil

Gabriel Poesia, Brasil

Jeferson Lesbão, Brasil

Maurício Collares, Brasil

Paulo Cezar Pereira Costa, Brasil

Ricardo Anido, Brasil

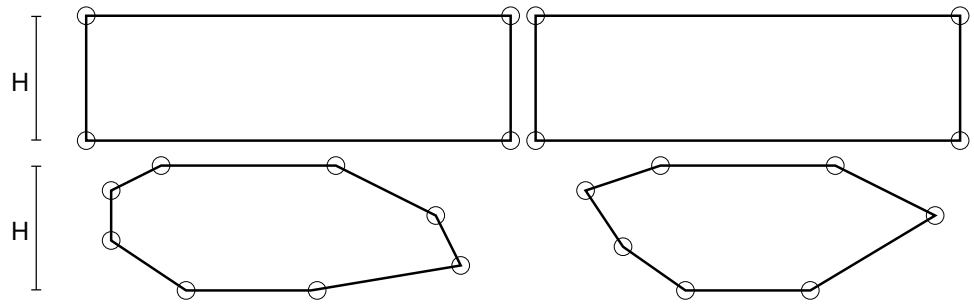
Problem A – Arranging tiles

Author: Guilherme A. Pinto, Brasil

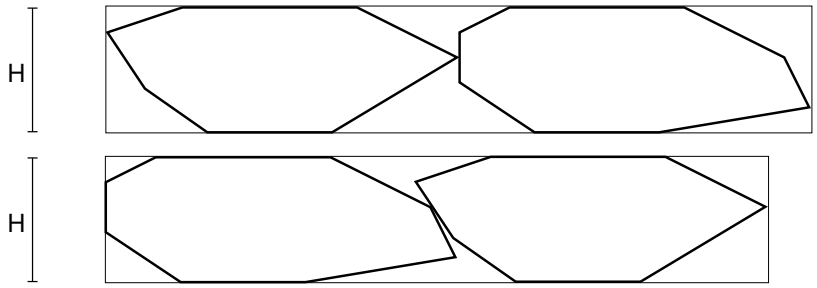
A set of rectangular stone tiles, all of them having the same height H, had their original four corners cut in different ways so that two properties were kept:

1. Each tile is still a simple convex polygon.
2. Each tile has two parallel sides that are part of the bottom and top sides of the original rectangular tile, which implies that the height H was preserved.

The figure below illustrates two tiles before and after the cuts. The corners are highlighted with small circles.



We need to place all tiles, side by side and without overlap, along a frame of height H, for transportation. The tiles can be translated from their original positions, but they may not be rotated or reflected. Since their convex shapes may be very different, the order in which we place the tiles along the frame matters, because we want to minimize the width of the frame. The next figure shows the two possible orders for the tiles from the previous figure, the second order being clearly the one that minimizes the width of the frame.



Given the description of the set of tiles, your program must compute the minimum width for a frame of the same height of the tiles that contains all of them, side by side and without overlap.

Input

The first line contains an integer N ($1 \leq N \leq 14$) representing the number of tiles. Following, there are N groups of lines, each group describing a tile, all of them having the same height.

Within each group describing a tile, the first line contains an integer K ($4 \leq K \leq 10^4$) representing the number of corners of the tile. Each of the next K lines describes a corner of the tile with two integers X ($-10^8 \leq X \leq 10^8$) and Y ($0 \leq Y \leq 10^8$), indicating the coordinates of the corner in the XY plane. The corners are given in counterclockwise order. The first corner is $(0, 0)$ and the second corner is of the form $(X, 0)$ for $X > 0$, this side being the bottom side of the tile. The tile has the shape of a simple convex polygon with a top side parallel to its bottom side.

Output

Output a single line with a rational number indicating the minimum width for a frame of the same height of the tiles that contains all of them, side by side and without overlap. The result must be output as a rational number with exactly three digits after the decimal point, rounded if necessary.

Sample input 1	Sample output 1
3 4 0 0 1 0 0 5 -1 5 4 0 0 1 0 2 5 1 5 4 0 0 3 0 2 5 1 5	5.000

Sample input 2	Sample output 2
3 4 0 0 204 0 412 1031 -253 1031 6 0 0 110 0 290 436 100 1031 0 1031 -400 750 5 0 0 120 0 100 1031 0 1031 -281 93	1420.754

Problem B – Buggy ICPC

Author: Gabriel Poesia, Brasil

Alan Curing is a famous sports programmer. He is the creator of the theoretical model of computation known as the Alan Curing Machine (ACM). He's most famous for creating his own computer for programming competitions: the Integrated Computer for Programming Contests (ICPC). This computer has a specialized operating system with commands for submitting code and testing executables on sample inputs, an input generator, a wide display for debugging, and a very soft keyboard. However, as it happens even to the best, Alan's creation has a nasty bug. Every time Alan types a vowel on the ICPC, the content of the current line is reversed.

The bug has been extremely hard to track down, so Alan has decided to accept the challenge and use the computer as it is. He is currently training touch typing on the ICPC. For now, he is only typing strings using lowercase letters, and no spaces. When Alan types a consonant, it is appended to the end of the current line, as one would expect. When he types a vowel, however, the typed character is first added to the end of the line, but right after that the whole line is reversed. For example, if the current line has "`imc`" and Alan types "`a`" (a vowel), for a brief moment the line will become "`imca`", but then the bug kicks in and turns the line into "`acmi`". If after that he types the consonants "`c`", "`p`" and "`c`", in that order, the line becomes "`acmicpc`".

When practicing, Alan first thinks of the text he wants to type, and then tries to come up with a sequence of characters he can type in order to obtain that text. He is having trouble, however, since he realized that he cannot obtain some texts at all (such as "`ca`"), and there are multiple ways of obtaining other texts (as "`ac`", which is obtained whether he types "`ac`" or "`ca`"). Help Alan in his training by telling him in how many ways he can type each text he wishes to type. A way of typing a text T can be encoded by a string W with $|T|$ characters such that if the characters are typed on the ICPC in the order they appear in W (i.e. $W_1, W_2, \ldots, W_{|T|}$) the final result is equal to T, considering ICPC's known bug. Two ways are considered different if they are encoded by different strings. The letters that trigger the bug in the ICPC when typed are "`a`", "`e`", "`i`", "`o`" and "`u`".

Input

The input consists of a single line that contains a non-empty string T of at most 10^5 lowercase letters, representing the text Alan wants to type on the ICPC.

Output

Output a single line with an integer representing the number of distinct ways Alan can type the desired text T considering ICPC's known bug.

Sample input 1	**Sample output 1**
`ac`	`2`

Sample input 2	**Sample output 2**
`ca`	`0`

Sample input 3	**Sample output 3**
`acmicpc`	`3`

This page would be intentionally left blank if we would not wish to inform about that.

Problem C – Complete Naebbirac's sequence

Author: Yonny Mondelo Hernández, Cuba

Naebbirac is a young and easy-to-get-bored sailor. He likes sequences of integers and to come up with ways to classify them. Naebbirac says that a sequence is *complete* for a chosen integer K, if the sequence only contains integers between 1 and K, and each integer between 1 and K appears the same number of times.

Based on that, Naebbirac created a game to entertain himself and his peers, when the waters calm down and there's not much they can do to spend their time in the middle of the ocean.

First he chooses a positive integer K and then he uses chalk to draw on the deck a sequence S having N integers between 1 and K. After that he challenges one of his peers. The goal of the challenged peer is to turn the sequence S into a *complete* sequence by performing exactly one of the following three possible operations:

- "`-x`": remove one occurrence of integer x from S;
- "`+x`": add a new integer with value x in S; or
- "`-x +y`": replace one occurrence of integer x from S by an integer with value y.

Naebbirac is quite smart. He never writes a sequence that is already *complete* and often the written integers don't follow a pattern, making it quite hard to find an operation that solves the puzzle. One of your friends, that usually sails with Naebbirac, is tired of always losing the game. Are you able to help your friend and create a computer program that can find a solution to Naebbirac's game before they go on their next trip?

Input

The first line contains two integers K ($3 \leq K \leq 1000$) and N ($1 \leq N \leq 10^4$), indicating respectively the integer that Naebbirac chooses at the beginning of the game, and the length of the sequence written on the deck. The second line contains N integers $S_1, S_2, \ldots, S_N$ ($1 \leq S_i \leq K$ for $i = 1, 2, \ldots, N$) representing the written sequence; you can safely assume that the sequence is not *complete*.

Output

Output a single line with the description of the operation that allows your friend to win the game or an "`*`" (asterisk) if there is no way to win. The description of the operation must follow the format shown on the statement, i.e. "`-x`", "`+x`" or "`-x +y`".

Sample input 1	**Sample output 1**
`3 5` `1 3 2 3 1`	`+2`

Sample input 2	**Sample output 2**
`3 7` `1 2 3 3 3 2 1`	`-3`

Sample input 3	**Sample output 3**
`3 6` `3 1 2 1 3 1`	`-1 +2`

Sample input 4	**Sample output 4**
`3 6` `2 3 2 2 2 1`	`*`

This page would be intentionally left blank if we would not wish to inform about that.

Problem D – Daunting device

Author: Walter Erquinigo, Perú

In a recent trip to an excavation site in the Caribbean island of Saint Basil, you found a mysterious device with some instructions resembling a puzzle. Your local guide Vibenas tells you that if you solve the puzzle, the device might show you the place where a big treasure left by the old merciless pirate Lyerpes is hidden.

The device has a tape with L cells indexed from 0 to $L-1$. Each cell has a color than can be changed with commands to the device. Each color is encoded by an integer, and initially all cells have the same color. The instructions that you found represent N steps to be performed before the device shows the way to the treasure. Each step is described using four integers P, X, A and B. The instructions say that to complete the step you must first count the number of cells currently having color P. Let this number be S. Then you must calculate the values

$$M_1 = (A + S^2) \mod L \ ,$$
$$M_2 = (A + (S + B)^2) \mod L \ .$$

Finally you have to make all cells within the closed interval $[\min(M_1, M_2), \max(M_1, M_2)]$ to be of color X.

After the exhausting task of processing the N steps required by the device, you still have one job: given a color that appears the greatest number of times in the device tape after all steps (that is, a most frequent color), you must go to the shipwreck of Lyerpes' legendary vessel and say aloud the number of cells having that color. Note that this number is unique even if more than one color appears the greatest number of times in the device tape after all steps.

Doing all those calculations on the device will take ages but you, as a renowned programmer, can create a program that quickly indicates the answer for the puzzle. After that, the real hard part of your mission will be to find out where is the shipwreck of Lyerpes' old vessel.

Input

The first line contains three integers L, C and N ($1 \leq L, C, N \leq 10^5$), representing respectively the number of cells in the tape, the number of available colors, and the number of steps in the instructions. Colors are identified by distinct integers from 1 to C and initially all cells have color 1. Each of the next N lines describes a step of the instructions with four integers P, X, A and B ($1 \leq P, X \leq C$ and $0 \leq A, B \leq 10^8$), indicating respectively the color whose number of cells is used to decide the range of the step, the color the cells in the range must have after the step is performed, and the other two values used to calculate the bounds of the range as described above.

Output

Given a color that appears the greatest number of times in the device tape after sequentially performing all steps described in the input, output a single line with an integer indicating the number of cells having that color.

Sample input 1	**Sample output 1**
7 5 2 1 2 5 3 3 3 0 1	4

Sample input 2	**Sample output 2**
7 10 8 10 6 5 6 5 1 7 5 9 9 10 1 3 2 6 7 8 3 4 8 3 7 7 4 9 3 9 7 1 1 8 1000	3

This page would be intentionally left blank if we would not wish to inform about that.

Problem E – Enigma

Author: Jeferson Lesbão, Brasil

The world famous pirate Cornelius "Cheesehead" Bakker was a renowned astronomer and mathematician. He buried most of his treasury in the Caribbean island of Saint Basil, where Pico Colombo is a well-known geographic reference mark. Cheesehead disappeared when his fleet of three ships was caught in a hurricane in 1617. Perhaps by some kind of premonition, before his fatal excursion he wrote in a letter to one of his nieces in the Netherlands the exact distance to his hidden treasure, from Pico Colombo in the south direction.

Wary that the map would end up in the wrong hands, Cheesehead used his math skills as an insurance against robbers. Instead of writing in the letter the number indicating the distance, he multiplied it by a second number N, and wrote the result D in the letter, together with the value of N and an explanation of the computation he had done. He knew that even if some unwanted person had the letter, he or she would have to know how to divide two numbers, which very few robbers could at that time. Unfortunately, when the letter arrived in Europe, Cheesehead's niece had joined a convent to become a nun and did not even bother to open the letter.

Exactly four centuries afterwards, Maria came into possession of a chest containing the belongings of her ancestor nun. And you can imagine her surprise when she found the letter, still unopened! Maria is planning an excursion to seek for Cheesehead's treasure, but she needs your help. Although the value of N is intact and she can read it, the number D has been partially eaten by moths so that some of its digits are unreadable. The only clue Maria has is that the leftmost digit of D is not zero because Cheesehead said so in the letter to his niece.

Given the partial representation of D and the value of N, you must determine the smallest possible value for D so that it is a multiple of N and does not have leading zeros.

Input

The input consists of a single line that contains a non-empty string S of at most 1000 characters and an integer N ($1 \leq N \leq 1000$). Each character of S is either a decimal digit or the character "`?`" (question mark); the leftmost character is not "`0`" and at least one character is "`?`".

Output

Output a single line with an integer D without leading zeros indicating the smallest multiple of N that has $|S|$ digits and such that the digits in S are coincident with the corresponding digits in D. If there exists no such an integer D, write an "`*`" (asterisk) to the output.

Sample input 1	**Sample output 1**
`1????????????????????????????????? 2`	`10000000000000000000000000000000`

Sample input 2	**Sample output 2**
`?????????????????????????????????1 2`	`*`

Sample input 3	**Sample output 3**
`?294?? 17`	`129404`

This page would be intentionally left blank if we would not wish to inform about that.

Problem F – Fundraising

Author: Paulo Cezar Pereira Costa, Brasil

A prestigious politician aiming for presidency next year is planning a fundraising dinner for her campaign. She has a list of some wealthy people in the country and wants to invite them in a way that the amount of money raised is as great as possible.

Sometimes wealthy people have futile behavior and don't like the idea that someone richer or prettier than them exists. Every time someone like this meets another person who is strictly prettier, but not strictly richer, then an argument ensues. Likewise, if they meet another person who is strictly richer, but not strictly prettier, an argument occurs as well. These two situations are the only possible causes of an argument involving two persons. Thus, two persons do not have an argument if one of them is strictly prettier and strictly richer than the other. Also, two persons do not have an argument if they are equally rich and equally pretty.

Since the presidential candidate wants to raise as much money as possible, an argument should be avoided at all costs, as it could ruin the campaign. Given the characteristics of some wealthy people in the country, you must find a guest list that maximizes the donations while ensuring that no argument will happen during the dinner.

Input

The first line contains an integer N ($1 \leq N \leq 10^5$) representing the number of possible guests with known characteristics. Each of the next N lines describes a possible guest with three integers B, F and D ($1 \leq B, F, D \leq 10^9$), indicating respectively the person's beauty, his/her fortune, and how much this person will donate if invited.

Output

Output a single line with an integer indicating the maximum sum of donations if guests are invited so that no argument will happen during the dinner.

Sample input 1	**Sample output 1**
4 1 2 50 2 1 50 2 2 30 1 1 30	60

Sample input 2	**Sample output 2**
3 3 3 3 5 5 3 2 2 3	9

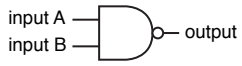
Sample input 3	**Sample output 3**
3 2 8 13 1 4 12 2 1 16	25

This page would be intentionally left blank if we would not wish to inform about that.

Problem G – Gates of uncertainty

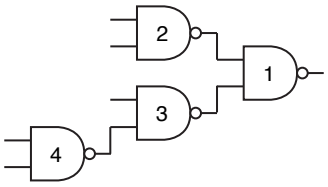
Author: Ricardo Anido, Brasil

A *NAND gate* (negative-AND gate) is a digital electronic circuit which produces an output that is false only if all its inputs are true; in other words, the output of a NAND gate is the complement to the output of an AND gate for the same inputs. A two-input NAND gate is a NAND gate with two inputs. The following figure shows the usual symbol of a two-input NAND gate and its truth table, using 1 for true and 0 for false.



input A	input B	output
0	0	1
0	1	1
1	0	1
1	1	0

In this problem we have a binary tree representing a circuit composed only by two-input NAND gates. In the tree, each internal node represents a NAND gate, which uses as inputs the values produced by its two children. Each leaf in the tree represents an *external input* to the circuit, and is a value in $\{0, 1\}$. The value produced by the circuit is the value produced by the gate at the root of the tree. The following picture shows a circuit with nine nodes, of which four are NAND gates and five are external inputs.



Each gate in the circuit may be *stuck*, meaning that it either only produce 0 or only produce 1, regardless of the gate's inputs. A *test pattern* is an assignment of values to the external inputs so that the value produced by the circuit is incorrect, due to the stuck gates.

Given the description of a circuit, you must write a program to determine the number of different test patterns for that circuit.

Input

The first line contains an integer N ($1 \leq N \leq 10^5$) representing the number of gates in the circuit, which has the shape of a binary tree. Gates are identified by distinct integers from 1 to N, gate 1 being the root of the tree. For $i = 1, 2, \ldots, N$, the i-th of the next N lines describes gate i with three integers X, Y and F ($0 \leq X, Y \leq N$ and $-1 \leq F \leq 1$). The values X and Y indicate the two inputs to the gate. If $X = 0$ the first input is from an external input, otherwise the input is the output produced by gate X. Analogously, if $Y = 0$ the second input is from an external input, otherwise the input is the output produced by gate Y. The value F represents the state of the gate: -1 means the gate is well-behaved, 0 means the gate is stuck at 0, and 1 means the gate is stuck at 1.

Output

Output a single line with an integer indicating the number of different test patterns for the given circuit. Because this number can be very large, output the remainder of dividing it by $10^9 + 7$.

Sample input 1	**Sample output 1**
4 2 3 1 0 0 -1 4 0 0 0 0 -1	15

Sample input 2	Sample output 2
2 2 0 1 0 0 -1	3

Sample input 3	Sample output 3
6 5 4 -1 0 0 -1 0 0 0 6 3 -1 0 2 1 0 0 -1	93

Sample input 4	Sample output 4
7 2 3 -1 4 5 -1 6 7 -1 0 0 1 0 0 -1 0 0 -1 0 0 -1	21

Problem H – Hard choice

Author: Inés Kereki, Uruguay

In long flights, airlines offer hot meals. Usually the flight attendants push carts containing the meals down along the aisles of the plane. When a cart reaches your row, you are asked right away: "Chicken, beef, or pasta?" You know your choices, but you have only a few seconds to choose and you don't know how your choice will look like because your neighbor hasn't opened his wrap yet...

The flight attendant in this flight decided to change the procedure. First she will ask all passengers what choice of meal they would prefer, and then she will check if the number of meals available in this flight for each choice are enough.

As an example, consider that the available number of meals for chicken, beef and pasta are respectively $(80, 20, 40)$, while the number of passenger's choices for chicken, beef and pasta are respectively $(45, 23, 48)$. In this case, eleven people will surely not receive their selection for a meal, since three passengers who wanted beef and eight passengers who wanted pasta cannot be pleased.

Given the quantity of meals available for each choice and the number of meals requested for each choice, could you please help the flight attendant to determine how many passengers will surely not receive their selection for a meal?

Input

The first line contains three integers C_a, B_a and P_a ($0 \leq C_a, B_a, P_a \leq 100$), representing respectively the number of meals available for chicken, beef and pasta. The second line contains three integers C_r, B_r and P_r ($0 \leq C_r, B_r, P_r \leq 100$), indicating respectively the number of meals requested for chicken, beef and pasta.

Output

Output a single line with an integer representing the number of passengers that will surely not receive their selection for a meal.

Sample input 1	**Sample output 1**
`80 20 40` `45 23 48`	`11`

Sample input 2	**Sample output 2**
`0 0 0` `100 100 100`	`300`

Sample input 3	**Sample output 3**
`41 42 43` `41 42 43`	`0`

This page would be intentionally left blank if we would not wish to inform about that.

Problem I – Imperial roads

Author: Edwin Niño, Colombia

The roads of Cubiconia are in a dire state, after years of neglect and lack of maintenance. Each road connects two different cities A and B and can be traveled in both ways (from A to B or from B to A). There is at most one road between each pair of cities, and using the existing roads it is possible to travel between any pair of cities. The new emperor of Cubiconia has just raised the taxes (again!), but promised to repair at least some of the roads, guaranteeing that Cubiconians will be able to travel between any pair of cities using only restored roads.

The Department of Public Works have calculated the cost of repairing each individual road. Now they want to calculate the minimum cost for repairing a set of roads so that the emperor's promise is made true. This is not easy because the emperor wants the set of repaired roads to include one particular road, but he has not yet decided which particular road to include: could be the one that connects the city where his castle is to the city where his daughter's royal residence is, or the road that connects the city where his summer palace is to the only city by the seaside, or... Fearing the emperor will take too long to decide, the engineers want your help.

Given the description of the roads in Cubiconia, with their respective repairing costs, you must write a program to answer a set of queries. For each query you will be given one specific road that should be repaired, and must determine the minimum cost for repairing a set of roads (including the given specific road) so that Cubiconians will be able to travel between any pair of cities using only restored roads.

Input

The first line contains two integers N ($2 \leq N \leq 10^5$) and R ($N - 1 \leq R \leq 2 \times 10^5$), representing respectively the number of cities and the number of roads in Cubiconia. Cities are identified by distinct integers from 1 to N. Each of the next R lines describes a road with three integers A, B ($1 \leq A < B \leq N$) and C ($1 \leq C \leq 10^4$), indicating that there is a road between cities A and B and the cost of repairing it is C. There is at most one road between each pair of cities, and using the existing roads it is possible to travel between any pair of cities. The next line contains an integer Q ($1 \leq Q \leq 10^5$) representing the number of queries. Each of the next Q lines describes a query with two integers U and V ($1 \leq U < V \leq N$), indicating the specific road that should be repaired. There are no repeated queries.

Output

Output Q lines, each line with an integer indicating the answer to the corresponding query of the input, that is, the minimum cost for repairing a set of roads (including the specific road in the query) so that Cubiconians will be able to travel between any pair of cities using only restored roads.

Sample input 1	**Sample output 1**
3 3 1 2 10 2 3 5 1 3 7 3 2 3 1 2 1 3	12 15 12

Sample input 2	**Sample output 2**
4 4 1 2 1 2 4 1 2 3 100 1 4 50 1 1 4	151

Sample input 3	**Sample output 3**
5 7 1 2 8 1 3 10 2 4 5 2 3 12 4 5 4 3 5 14 1 5 20 3 2 3 1 5 3 5	29 39 31

Problem J – Jumping Frog

Author: Gabriel Poesia, Brasil

Pog the Frog wants to compete in the World Frog Jump competition, which will take place in Nlogonia. In the competition, each frog must perform a sequence of acrobatic jumps in a specially built arena. The arena is composed of N equally spaced positions around a circumference (the arc between two adjacent positions is always the same length) where each position can be either a rock or a pond. The positions are numbered sequentially from 0 to $N-1$ in the clockwise direction, so that judges can easily make notes about which jumps were performed in each position. Thus, position 0 is adjacent to positions 1 and $N-1$ in the arena.

The competition rules stipulate that the sequence of jumps of each frog must start at a rock, always go from a rock to another rock, and finish at the same position it started. The rules do not require frogs to use every rock in the arena for their sequence of jumps.

Pog the Frog is currently practicing for the competition. He must develop two skills. First, he needs to get better at jumping from one rock to another, since landing on either a pond or outside of the marked positions can mean disqualification. Besides that, he must learn impressing acrobatic moves. With that in mind, he has decided on a practicing strategy. In the beginning of each practice session, Pog the Frog will pick a starting rock and an integer *jump length* K between 1 and $N-1$. After that, whenever he is standing on a rock numbered i, he will aim his next acrobatic jump at the rock whose number is obtained by getting the remainder of the division of $i+K$ by N. He will stop when he lands on the starting rock. For example, if the arena has 3 positions, all of them rocks, and Pog the Frog starts at position 0 and picks $K=2$, he will first jump from rock 0 to rock 2, then to rock 1, and finally jump back to rock 0. At this point, his practice session ends.

Given the description of the N positions in the arena, help Pog the Frog by answering this question: how many distinct values of K can he choose for his practice sessions, if he can use any rock as a starting position for his sequence of jumps?

Input

The input consists of a single line that contains a string S of N characters ($3 \leq N \leq 10^5$), representing the positions in the arena. The i-th character of S ($i = 0, 1, \ldots, N-1$) indicates that the position i in the arena is either a rock (uppercase letter "`R`") or a pond (uppercase letter "`P`").

Output

Output a single line with an integer representing the number of distinct jump lengths that Pog the Frog can choose for his practice sessions, given that he can use any rock as a starting position for his sequence of jumps.

Sample input 1	**Sample output 1**
`RRR`	`2`

Sample input 2	**Sample output 2**
`RRPR`	`1`

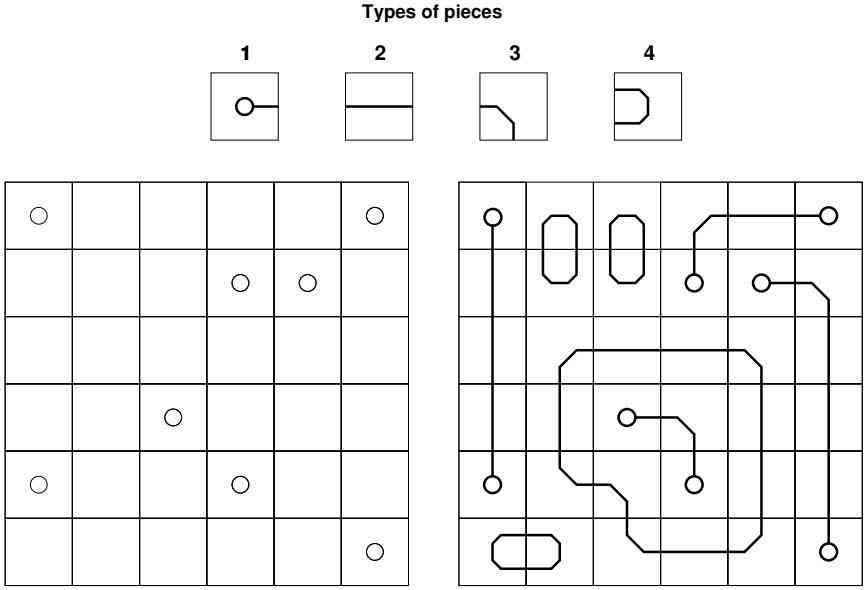
Sample input 3	**Sample output 3**
`PRP`	`0`

This page would be intentionally left blank if we would not wish to inform about that.

Problem K – Keep it covered

Author: Paulo Cezar Pereira Costa, Brasil

Eve loves puzzles. She recently bought a new one that has proven to be quite difficult. The puzzle is made of a rectangular grid with R rows and C columns. Some cells may be marked with a dot, while the other cells are empty. Four types of pieces come with the puzzle, and there are $R \times C$ units of each type.



The objective of the puzzle is to use some of the pieces to completely fill the grid; that is, each cell must be covered with a piece. In doing that, each piece may be rotated 90, 180 or 270 degrees. But of course, to make it more interesting, there are a few constraints that must be respected:

1. Type 1 pieces can only be used on cells marked with a dot, while the other types of pieces can only be used on empty cells.
2. Given any pair of cells sharing an edge, the line drawings of the two pieces on them must match.
3. The line drawings of the pieces cannot touch the border of the grid.

As Eve is having a hard time to solve the puzzle, she started thinking that it was sloppily built and perhaps no solution exists. Can you tell her whether the puzzle can be solved?

Input

The first line contains two integers R and C ($1 \leq R, C \leq 20$), indicating respectively the number of rows and columns on the puzzle. The following R lines contain a string of C characters each, representing the puzzle's grid; in these strings, a lowercase letter "`o`" indicates a cell marked with a dot, while a "`-`" (hyphen) denotes an empty cell. There are at most 15 cells marked with a dot.

Output

Output a single line with the uppercase letter "`Y`" if it's possible to solve the puzzle as described in the statement, and the uppercase letter "`N`" otherwise.

Sample input 1	**Sample output 1**
6 6 o----o ---oo- ------ --o--- o--o-- -----o	Y

Sample input 2	Sample output 2
1 1 -	N

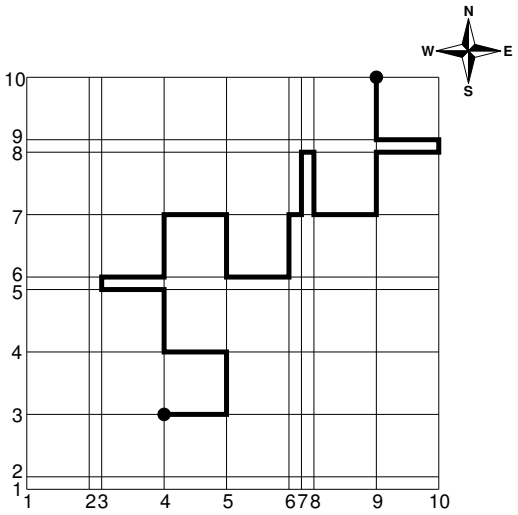
Sample input 3	Sample output 3
6 7 ------- -o--o-- --o---- -----o- ----o-- o------	N

Sample input 4	Sample output 4
3 3 -o- o-o -o-	N

Problem L – Linearville

Author: Guilherme A. Pinto, Brasil

The city of Linearville has N parallel two-way streets going in the West-East direction and N parallel two-way streets going in the South-North direction, making up a grid with $(N-1) \times (N-1)$ blocks. The distance between two consecutive parallel streets is either 1 or 5. The Linearville Transit Authority is conducting an experiment and now requires all cars to always follow a path that alternates direction between W-E and S-N at every crossing, meaning they must turn either left or right when reaching a crossing. The LTA is developing a new navigation app and needs your help to write a program to compute the lengths of shortest alternating paths between many pairs of start and target crossings. The alternating path in the figure, as an example for $N = 10$, is clearly not a shortest alternating path. But beware! Linearville may be huge.



Input

The first line contains an integer N ($2 \leq N \leq 10^5$) representing the number of streets in each direction. For each direction, the streets are identified by distinct integers from 1 to N starting at the S-W corner of the city. The second line contains $N-1$ integers $D_1, D_2, \ldots, D_{N-1}$ ($D_i \in \{1, 5\}$ for $i = 1, 2, \ldots, N-1$) indicating the distances between consecutive streets going S-N (that is, D_i is the distance between street i and street $i+1$). The third line contains $N-1$ integers $E_1, E_2, \ldots, E_{N-1}$ ($E_i \in \{1, 5\}$ for $i = 1, 2, \ldots, N-1$) indicating the distances between consecutive streets going W-E (that is, E_i is the distance between street i and street $i+1$). The fourth line contains an integer Q ($1 \leq Q \leq 10^5$) representing the number of shortest path queries. Each of the next Q lines describes a query with four integers A_X, A_Y, B_X and B_Y ($1 \leq A_X, A_Y, B_X, B_Y \leq N$), indicating that the start crossing is (A_X, A_Y) and the target crossing is (B_X, B_Y); the values A_X and B_X are streets going S-N while the values A_Y and B_Y are streets going W-E. There are no repeated queries.

Output

Output Q lines, each line with an integer indicating the length of a shortest alternating path for the corresponding query of the input.

Sample input 1	**Sample output 1**
10 5 1 5 5 5 1 1 5 5 1 5 5 5 1 5 5 1 5 3 4 3 9 10 9 2 2 9 5 1 5 10	46 50 49

Sample input 2	**Sample output 2**
5 5 1 5 5 5 1 5 5 2 3 1 4 5 5 5 5 5	23 0

This page would be intentionally left blank if we would not wish to inform about that.

Problem M – Marblecoin

Author: Paulo Cezar Pereira Costa, Brasil

Cubiconia is known for having one of the highest tax rates. Taxes are calculated on a daily basis and even things that seem worthless are subject to taxes. In order to avoid the crazy tax rates, some of the emperor's friends created a new currency using marbles. Unfortunately, it didn't work out, marbles became subject to taxes as well.

Despite this, the emperor believes that using marbles as a currency is a great idea and that in the future they will be worth a lot more. So he decided to steal all of his friends' marbles. To avoid unnecessary attention, in the dead of each night he will visit one of his friends and during each visit exactly one marble will be stolen. Since the emperor's friends keep their marbles in stacks, only a marble that is currently on the top of a stack might be stolen.

Each marble has a value associated to it. The amount due per owned marble is $V \times 365^D$, where V is the value associated to the marble and D is the number of days the marble was owned. The emperor plans to sell all the marbles once he is finished stealing them. This means that, if there is a total of T marbles, the marble owned the least amount of time will be owned for 1 day, while the one owned the maximum amount of time will be owned for T days.

The emperor is smart and already realized that the total due in taxes depends on the order in which marbles are stolen. To avoid paying more taxes than necessary, he would like to know the best order to steal the marbles. Can you help him?

Input

The first line contains an integer N ($1 \leq N \leq 10^5$) representing the number of stacks the emperor is going to steal from. Each of the next N lines describes a stack with an integer K ($1 \leq K \leq 10^5$) followed by K integers $V_1, V_2, \ldots, V_K$ ($1 \leq V_i \leq 300$ for $i = 1, 2, \ldots, K$); the number K is the amount of marbles in the stack, while the numbers $V_1, V_2, \ldots, V_K$ are the values of the marbles in the stack, from top to bottom. The total amount of marbles is at most 4×10^5.

Output

Output a single line with an integer representing the minimum value due in taxes if the marbles are stolen in an optimal order. Because this number can be very large, output the remainder of dividing it by $10^9 + 7$.

Sample input 1	**Sample output 1**
3 1 1 1 2 1 3	48894670

Sample input 2	**Sample output 2**
3 3 2 5 7 4 1 4 6 10 3 3 2 1	227712621

This page would be intentionally left blank if we would not wish to inform about that.