International Collegiate Programming Contest

The 2025 ICPC Latin America Championship

Problem Set

Contest Session

March 16, 2025

icpc.foundation

icpc Latin America Championship

This problem set contains 12 problems; pages are numbered from 1 to 27.

v1.0

JETBRAINS

icpc global sponsor programming tools

HUAWEI

icpc diamond multi-regional sponsor

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*`.kt`, *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.

Problem A – Ananna

Danaland is a very typical country: it consists of N cities, each identified by a distinct number. These cities are connected by M unidirectional roads, where each road has a name.

Ananna is a bright little girl who lives in Danaland. Unfortunately, she was born with a terrible disease: she can only read backwards. After being a victim of terrible bullying by her peers, or, as Ananna calls them, sreep, she found solace in palindromes: words that are the same when read backwards.

Ananna's mom, Eeve, is trying to help her with her condition. One way she helps is by taking her on road trips. A road trip is a sequence of roads that starts at some city U and ends at a different city V; the same road may appear more than once.

While on a road trip, Eeve asks Ananna the first letter of each road name, so she can practice looking at the start of words. This is, obviously, a source of great anxiety to Ananna, so to avoid having a kcatta cinap, Eeve always makes sure that the sequence formed by taking the first letter of each road's name, in the order they are traversed, is a palindrome.

Eeve is now looking at a map of Danaland, and she wonders: How many distinct pairs of cities U, V exist such that Eeve can take a road trip from U to V?

Input

The first line contains two integers N and M ($1 \leq N, M \leq 5000$), indicating respectively the number of cities and the number of roads in Danaland. Each city is identified by a distinct integer from 1 to N.

Each of the next M lines contains two integers U and V ($1 \leq U, V \leq N$) and a lowercase letter C, representing that there is a unidirectional road from U to V whose name starts with C. Several roads may connect the same pair of cities, and a road may connect a city to itself.

Output

Output a single line with an integer indicating the number of pairs of cities U, V such that $U \neq V$, there is a road trip from U to V, and the letters of the roads (in the order they are traversed) form a palindrome.

Sample input 1	Sample output 1
4 6 1 2 b 2 3 a 3 4 a 1 1 a 4 3 d 4 3 c	7

Explanation of sample 1:
The 7 pairs of cities and possible road trips are:

- $1, 2 : 1 \xrightarrow{b} 2$
- $1, 3 : 1 \xrightarrow{a} 1 \xrightarrow{b} 2 \xrightarrow{a} 3$
- $1, 4 : 1 \xrightarrow{a} 1 \xrightarrow{a} 1 \xrightarrow{b} 2 \xrightarrow{a} 3 \xrightarrow{a} 4$
- $2, 3 : 2 \xrightarrow{a} 3$
- $2, 4 : 2 \xrightarrow{a} 3 \xrightarrow{a} 4$
- $3, 4 : 3 \xrightarrow{a} 4$
- $4, 3 : 4 \xrightarrow{d} 3$

Sample input 2	Sample output 2
2 3 1 1 x 2 2 y 1 1 z	0

Problem B – Brazilian FootXOR

Alex is coaching the famous Brazilian FootXOR Club. For today's training, they want to organize a match between two teams selected from the N club members. To ensure a fair match, the two teams must be balanced.

Each club member has a set of abilities represented as a K-digit binary string. Each digit corresponds to a characteristic that the player may or may not have. Surprisingly, it is not the quantity of players having each characteristic that matters when deciding whether two teams are balanced: the important thing is the parity of that quantity.

Two teams are considered balanced if they have the same number of players and the bitwise XOR of the ability strings of all players in the first team is equal to the bitwise XOR of the ability strings of all players in the second team. Of course, each club member can only be assigned to at most one team and, for a match to happen, teams cannot be empty.

Alex is really busy right now. As their coaching assistant, you must determine how to assign players to the two teams. If it is impossible to fulfill the coach's conditions, you must inform the coach accordingly.

Input

The first line contains two integers N and K ($1 \leq N, K \leq 1500$), indicating respectively the number of club members and the number of binary digits used to represent their abilities. Each club member is identified by a distinct integer from 1 to N.

The i-th of the next N lines contains a K-digit binary string A_i, representing the abilities of club member i. Note that A_i has a fixed length and so it may contain leading zeros.

Output

Output a single line with a string of length N if it is possible to fulfill the coach's conditions. In this case, the i-th character of the string must be a digit "1", "2" or "0", indicating respectively that club member i is assigned to the first team, to the second team, or to none of the two teams. If there are multiple solutions, output any of them.

If the coach's conditions cannot be honored, output the character "*" (asterisk) instead.

Sample input 1	Sample output 1
5 3 101 001 010 011 000	01221

Sample input 2	Sample output 2
3 2 01 10 11	*

Sample input 3	Sample output 3
4 3 011 000 100 011	1002

Sample input 4	Sample output 4
1 3 000	*

Problem C – Coatless in Yakutsk

It was your first time visiting Salvador, and you made the rookie mistake of sleeping on the beach. You woke up red, sunburned, and frankly, humiliated. Swearing vengeance against the sun and all its terrible consequences, you decided that your next vacation would be somewhere with a real winter – like Yakutsk, Russia, where the average temperature is $-42°$ Celsius.

But you came prepared! You brought a warm, cozy coat. The coat is perfect, it warms you very well. Maybe too well, as you get sweaty and the coat gets dirty after C days of use. Since your trip lasts more than C days, you must find a way to avoid walking around smelling bad.

To do so, when the coat gets dirty, you cannot wear it until it is washed, but you may also choose to wash it earlier. On any day you do not wear the coat – whether because it is dirty or being washed – you must endure the day's temperature without its protection. After being washed, the coat is fresh and ready to be worn again. At the start of your trip, the coat is clean.

Given the daily temperatures in Yakutsk for the duration of your trip, you must determine the lowest temperature on a day when you are forced to be without your coat, assuming you schedule wash days optimally to make this temperature as high as possible.

Input

The first line contains two integers C and N ($1 \leq C < N \leq 10^5$), indicating respectively the number of days you can wear the coat before it gets dirty, and the duration in days of your holidays.

The second line contains N integers $T_1, T_2, \ldots, T_N$ ($-50 \leq T_i \leq 50$ for $i = 1, 2, \ldots, N$), where T_i is the temperature on the i-th day.

Output

Output a single line with an integer indicating the minimum temperature you must endure without your coat.

Sample input 1	Sample output 1
2 6 -20 -10 -5 -10 -2 -40	-5

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

Problem D – Dangerous City

Alice is moving to the city of Nlogonia, and to decide where to live, she is evaluating the safety of the city.

Nlogonia is a planned city with N intersections, numbered from 1 to N, and M streets. Each street connects two intersections bidirectionally. It is guaranteed that any intersection can reach all other intersections using the streets, and no two streets connect the same pair of intersections.

The government of Nlogonia publishes a **danger rating** D_i for each intersection i. However, Alice thinks these ratings are insufficient because she wants to assess the safety of moving through the city, not just where she lives. So, she developed her own way to measure how dangerous the city is.

For any given path in the city, Alice defines its **path risk** as the **maximum** danger rating among all intersections on that path, including its endpoints. The **risk factor** between two intersections U and V, denoted as $f(U, V)$, is the **minimum** possible path risk among all paths connecting U and V. By definition, the only path from an intersection U to itself is the trivial path containing only U, so we have $f(U, U) = D_U$. Finally, she assigns a **danger score** to each intersection U, denoted as:

$$S_U = \sum_{V=1}^{N} f(U, V)$$

In other words, the danger score of an intersection U is the sum of its risk factors to every intersection in the city.

Computing these danger scores for all intersections is not easy, so Alice asks for your help!

Input

The first line contains two integers N $(2 \le N \le 3 \cdot 10^5)$ and M $(1 \le M \le 3 \cdot 10^5)$, indicating respectively the number of intersections and streets in Nlogonia. Each intersection is identified by a distinct integer from 1 to N.

The second line contains N integers $D_1, D_2, \ldots, D_N$ $(1 \le D_i \le 10^9$ for $i = 1, 2, \ldots, N)$, where D_i is the danger rating of intersection i.

Each of the next M lines contains two integers U and V $(1 \le U, V \le N$ and $U \ne V)$, indicating that there is a two-way street between intersections U and V.

It is guaranteed that there is at most one street between each pair of intersections and that any intersection can be reached from any other using one or more streets.

Output

Output a single line with N integers $S_1, S_2, \ldots, S_N$, that is, the danger scores of all the intersections.

Sample input 1	Sample output 1
3 2 1 3 1 1 2 2 3	7 9 7

Sample input 2	Sample output 2
3 3 1 3 1 2 3 1 2 3 1	5 9 5

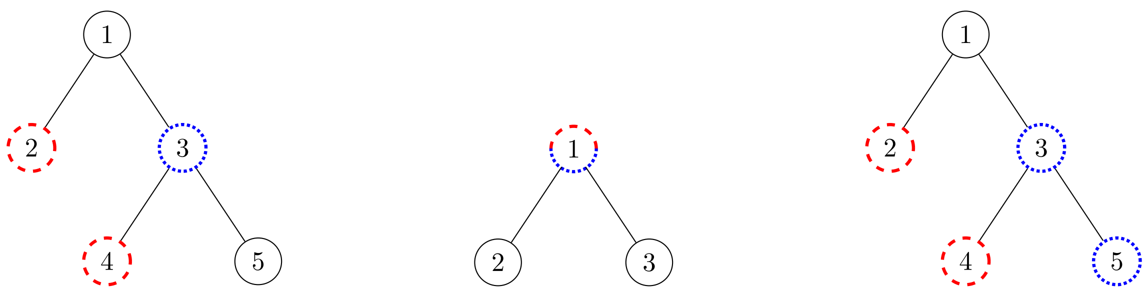
Problem E – Exciting Business Opportunities

The Treeland subway system consists of N stations connected by $N - 1$ bidirectional tunnels. The tunnels are arranged in such a way that it is always possible to travel between any two stations. The system is already considered to be the most efficient in the country: there are trains between every pair of stations in the system.

Alejandro, from the subway improvement department, has been tasked with making the system even better and more attractive for passengers and businesses. To that end, the department collected P proposals from companies. Each proposal is either to sponsor some station (basically, add a sign with the name of the company on the station), or to open a business at some station (a restaurant, a store, or anything like this). Notice that there is no restriction on the number of proposals that can be accepted for each station.

Companies willing to open businesses in the system, though, informed that they will withdraw their proposals unless the station X where their business is located, is either a sponsored station or there is a train between two sponsored stations that goes through X. Of course, Alejandro does not want to select proposals that will later be withdrawn. Thus, he will make sure to select a set of proposals such that no proposal in the set will be withdrawn. He calls such a set a *valid* set of proposals.

The figure below shows sponsored stations in red/dashed circles and businesses in blue/dotted circles. The set of proposals in (a) is valid because the business proposal for station 3 is located exactly on the path between the two sponsored stations 2 and 4. The set of proposals in (b) is also valid, because the business proposal for station 1 is located exactly on a sponsored station. On the contrary, the set of proposals in (c) is not valid: even though the business proposal for station 3 is located on a path between two sponsored stations, the business proposal for station 5 is not.



(a) Valid (b) Valid (c) Not valid

To choose a valid set, Alejandro took the P proposals (each one described on a piece of paper), and numbered them from 1 to P. Now he wants to select a contiguous range of proposals that form a valid set. More precisely, he wants to pick two integers i and j, with $1 \leq i \leq j \leq P$, such that proposals $i, i+1, \ldots, j$ form a valid set; besides, he wants the set to be as large as possible. Alejandro is not sure about which should be the initial proposal i in the range he will select. Help him compute for each i from 1 to P, the size of the largest contiguous range of proposals that form a valid set, starting at proposal i.

Input

The first line contains an integer N ($2 \leq N \leq 10^5$) indicating the number of stations in the subway system. Each station is identified by a distinct integer from 1 to N.

Each of the next $N - 1$ lines contains two integers U and V ($1 \leq U, V \leq N$ and $U \neq V$), representing that there is a bidirectional tunnel between stations U and V. It is guaranteed that it is possible to go from any station to any other station using the tunnels.

The next line contains an integer P ($1 \leq P \leq 10^5$) denoting the number of proposals submitted by companies. Each proposal is identified by a distinct integer from 1 to P.

The i-th of the next P lines describes proposal i with a character C and an integer X ($1 \leq X \leq N$). For a sponsoring proposal C is the lowercase letter "`s`", while for a business proposal it is the lowercase letter "`b`"; in both cases X is the corresponding station. Notice that each station can have an arbitrary number of proposals of any type.

Output

Output P lines. The i-th line must contain an integer indicating the size of the largest valid set of contiguous proposals whose initial proposal is i.

Sample input 1

```
6
2 1
1 3
3 5
5 6
4 3
8
b 1
s 2
b 6
s 3
b 4
s 5
b 3
s 4
```

Sample output 1

```
0
1
0
5
4
3
0
1
```

Sample input 2

```
7
4 2
2 1
1 3
3 6
5 2
3 7
6
s 4
b 5
b 6
s 7
b 2
s 3
```

Sample output 2

```
1
0
0
1
0
1
```

Sample input 3	Sample output 3
2 1 2 5 s 1 b 1 s 1 b 1 s 1	5 4 3 2 1

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

Problem F – Festival Signs

During a festival, advertisement signs are added to and removed from a stage at different moments. Each sign has a rectangular shape and is placed perpendicular to the ground, with one side resting firmly on the ground.

A webcam is livestreaming the festival, showing a 2D image of the stage. In this image, the bottom border corresponds to the ground. Each sign covers a rectangular area in the image. A sign is described by an interval $[A, B]$ and a height H, meaning it covers all points (x, y) of the image where $A \leq x \leq B$ and $0 \leq y < H$.

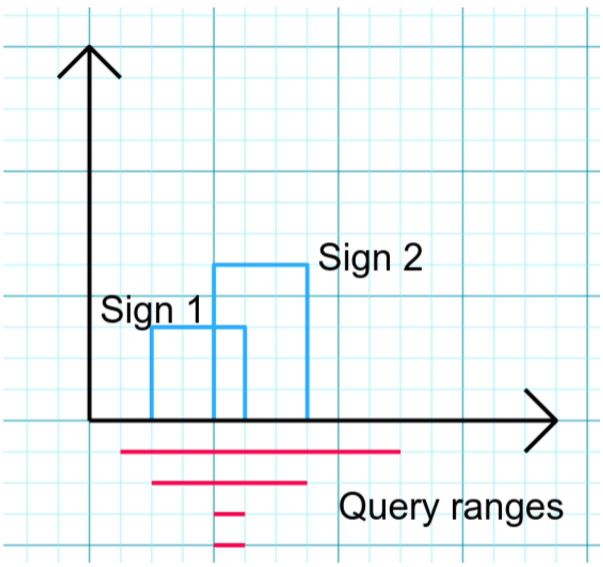
Note that the sign covers the points on the bottom and side borders of the rectangle, but not the top border. Besides, rectangles representing different signs may overlap.

During the livestream, the manager will make several queries at different moments. Each query specifies an interval $[L, R]$, and asks for the minimum height $y \geq 0$ among all uncovered points (x, y) with $L \leq x \leq R$, with x and y being real numbers.

Your task is to write a program to process a sequence of N events: sign additions, sign removals, and queries about the minimum uncovered height over a given interval.

As an example, consider the following sequence of $N = 7$ events given in chronological order (see picture below):

- Sign with $[A, B] = [2, 5]$ and $H = 3$ is added.
- Sign with $[A, B] = [4, 7]$ and $H = 5$ is added.
- Query $[1, 10]$ is made. The answer to this query is 0. An uncovered point with minimum height within the interval is, for instance, the point with coordinates $(1.5, 0)$.
- Query $[2, 7]$ is made, with answer 3.
- Query $[4, 5]$ is made, with answer 5.
- Second added sign is removed.
- Query $[4, 5]$ is made, with answer 3. Note that the removal of the second added sign changed the answer of query $[4, 5]$.



Input

The first line contains an integer N ($1 \leq N \leq 2 \cdot 10^5$) indicating the number of events.

Each of the next N lines describes an event, in chronological order. The content of the line depends on the event, as follows:

- Sign addition: the line contains the character "+" (plus sign) and three integers A, B and H ($1 \leq A, B, H \leq 10^9$ and $A < B$), representing that a sign is added, covering a rectangle of the image as explained in the statement. Each added sign is assigned a sequential integer identifier, starting from 1.
- Sign removal: the line contains the character "-" (minus sign) and an integer $I \geq 1$, indicating that the sign with identifier I is removed. It is guaranteed that I identifies an added and not previously removed sign.
- Query: the line contains the character "?" (question mark) and two integers L and R ($1 \leq L < R \leq 10^9$), asking for the minimum uncovered height within the interval $[L, R]$, as explained in the statement. It is guaranteed that the input contains at least one query.

Output

Output a line for each query, with an integer indicating the minimum uncovered height within the corresponding interval.

Sample input 1	Sample output 1
7 + 2 5 3 + 4 7 5 ? 1 10 ? 2 7 ? 4 5 - 2 ? 4 5	0 3 5 3

Sample input 2	Sample output 2
4 + 1 2 1 + 3 4 1 ? 1 2 ? 1 4	1 0

Problem G – Game of Pieces

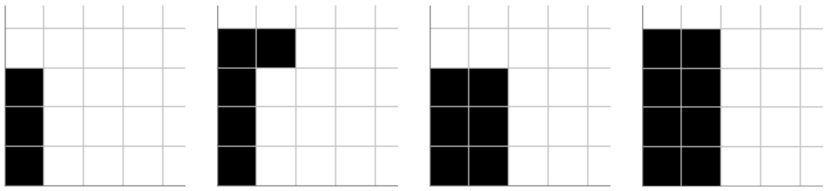
If you've ever played Tetris for long enough, you might have experienced the *Tetris effect* – seeing falling blocks even after you have stopped playing. To stay focused on solving problems and avoid such distractions, we will consider a simplified version of the game.

The game is played on a board composed of square cells arranged in a grid. Columns of the grid are numbered sequentially from left to right. The board is infinite to the right and to the top. Each cell is either empty or filled, and initially, all cells are empty.

A sequence of N rectangular pieces is given, and the pieces are dropped onto the board one at a time. Pieces have different sizes. A piece of size L is either vertical ($L \times 1$ cells) or horizontal ($1 \times L$ cells). When a piece is dropped at a specified column, it starts at a location above all the currently filled cells and falls straight down until it either reaches the bottom of the board or lands on top of an already filled cell. Once a piece lands, it fills its final set of cells.

Each time a piece lands, the game is considered *safe* if no empty cell has a filled cell above it; otherwise the game is *unsafe*, the offending piece is removed from the board, and the game continues with the next piece as if it had never been dropped.

In the example below, corresponding to the first sample input, the game becomes unsafe once the second piece lands, hence the piece is removed and the next pieces keep the game safe.



Given the sequence of pieces and their drop positions, your task is to determine, for each piece, whether it makes the board unsafe after landing.

Input

The first line contains an integer N ($1 \leq N \leq 2 \cdot 10^5$), indicating the number of pieces.

Each of the next N lines describes a piece with a character C and two integers L and P ($1 \leq L \leq 10^9$ and $1 \leq P \leq 10^{18}$), representing respectively the type of the piece, its length, and the position where it is dropped. For a vertical piece, C is the character "`|`" (pipe), the piece has $L \times 1$ cells, and it is dropped at column P. For a horizontal piece, C is the character "`-`" (hyphen), the piece has $1 \times L$ cells, and it is dropped spanning columns $P, P+1, \ldots, P+L-1$.

Output

Output a single line with a string of length N. The i-th character of the string must be the uppercase letter "`U`" if the game becomes unsafe immediately after the i-th piece lands on the board, and the uppercase letter "`S`" otherwise.

Sample input 1	Sample output 1

`4`
`| 3 1`
`- 2 1`
`| 3 2`
`- 2 1`
	`SUSS`

Sample input 2	Sample output 2
4 | 3 1 | 2 3 | 1 2 - 2 2	SSSU

Problem H – Horrible Restaurants

Ricardo is a restaurant critic, which means he spends his time eating at restaurants and giving them a rating. Each rating is an integer number of stars between 0 and 3, inclusive. Thus, there are exactly four possible ratings.

During his visit to Cheapland, he must review N restaurants. Unfortunately, all of them are terrible, and if left to his honest opinion, Ricardo would give 0 stars to every restaurant. However, the government of Cheapland can bribe Ricardo to increase the rating of any restaurant.

Each restaurant has its own bribe costs, which depend both on the restaurant itself and on the number of stars awarded. Bribing Ricardo to give a restaurant 3 stars is always more expensive than bribing him for 2 stars, which in turn is more expensive than bribing him for 1 star. Naturally, no payment is required for a 0-star rating.

As one might imagine, the Cheapland government wants to spend as little as possible while making their gastronomic scene look as strong as possible. To plan their strategy, they need to determine the minimum cost required to achieve a total of k stars among all the N restaurants, for every integer value k between 1 and $3N$, inclusive.

However, since bribe costs vary from restaurant to restaurant, calculating these values isn't straightforward – which is why they need your help.

Input

The first line contains an integer N ($1 \leq N \leq 2 \cdot 10^5$) indicating the number of restaurants.

Each of the next N lines describes a restaurant with three integers C_1, C_2 and C_3 ($1 \leq C_1 < C_2 < C_3 \leq 10^9$), where C_i is the cost of getting a rating of i stars for the restaurant.

Output

Output a line for each k from 1 to $3N$ (inclusive), with an integer indicating the minimum total cost required to achieve exactly k stars among all the N restaurants.

Sample input 1	Sample output 1
3 1 2 3 2 10 11 5 6 7	1 2 3 5 9 10 12 20 21

Sample input 2	Sample output 2
4 999999998 999999999 1000000000 999999998 999999999 1000000000 999999998 999999999 1000000000 999999998 999999999 1000000000	999999998 999999999 1000000000 1999999998 1999999999 2000000000 2999999998 2999999999 3000000000 3999999998 3999999999 4000000000

Problem I – Infinite Arrays

A subarray of an array B is a contiguous sequence of elements taken from B. For example, if $B = [3, 1, 4, 1, 5]$, then $[3, 1, 4]$, $[4, 1]$, the empty array $[\ ]$ and the whole array B are subarrays of B (among others), while $[3, 4, 5]$ and $[1, 3]$ are not subarrays of B.

We also define C^m as the array obtained by the concatenation of m copies of the array C. For example, if $C = [3, 2]$ then $C^1 = [3, 2]$, $C^2 = [3, 2, 3, 2]$ and $C^\infty = [\ldots, 3, 2, 3, 2, 3, 2, \ldots]$.

In this problem you are given an array P containing no repeated numbers, and you have to process a sequence of events that occur in order. The events can be of three types:

- Delete: a number present in P must be deleted. For example, if $P = [2, 3, 4]$ and the number 3 has to be deleted, once the event is processed P will become $[2, 4]$.
- Insert: a number not present in P must be inserted into P before some other number present in P. For example, if $P = [4, 3, 2, 1]$ and the number 9 must be inserted before the number 1, once the event is processed P will become $[4, 3, 2, 9, 1]$.
- Query: the length of the longest common subarray between P^∞ and A^∞ must be computed, where A is an array given in the query. For example, if $P = [1, 2, 3, 4]$ and $A = [3, 2]$, then the longest common subarray between P^∞ and A^∞ is $[2, 3]$, so the answer to the query is 2.

Are you ready for this challenge?

Input

The first line contains an integer N ($1 \le N \le 5 \cdot 10^5$) indicating the initial length of P.

The second line contains N different integers $P_1, P_2, \ldots, P_N$ ($1 \le P_i \le 10^6$ for $i = 1, 2, \ldots, N$).

The third line contains an integer E ($1 \le E \le 5 \cdot 10^5$) representing the number of events that need to be processed.

Each of the next E lines describes an event, in the order they must be processed. The content of the line depends on the event, as follows:

- Delete: the line contains the character "-" (minus sign) and an integer X ($1 \le X \le 10^6$, and $X \in P$), denoting that X must be deleted from P. It is guaranteed that after the removal P does not become empty.
- Insert: the line contains the character "+" (plus sign) and two integers Y and Z ($1 \le Y, Z \le 10^6$, $Y \notin P$, and $Z \in P$), denoting that Y must be inserted into P immediately to the left of Z.
- Query: the line contains the character "?" (question mark), a positive integer K, and K integers $A_1, A_2, \ldots, A_K$ ($1 \le A_i \le 10^6$ for $i = 1, 2, \ldots, K$), indicating that the length of the longest common subarray between P^∞ and A^∞ must be computed. It is guaranteed that the input contains at least one query, and the sum of K across all the queries is at most 10^6.

Output

Output a line for each query, with an integer indicating the length of the longest common subarray between P^∞ and A^∞, or the character "*" (asterisk) if the length of the longest common subarray is larger than 10^{18}.

Sample input 1	Sample output 1
4 1 2 3 4 10 ? 2 3 2 ? 3 99 99 99 - 1 - 4 ? 2 3 3 ? 3 4 1 4 + 64 3 + 1 2 ? 4 1 2 64 3 ? 4 1 3 64 2	2 0 1 0 * 1

Sample input 2	Sample output 2
1 217 1 ? 1 314	0

Problem J – Just Look Up

The year is 2432, and after millennia of looking at the sky, astronomers have found all the stars. This arduous journey, which started with the discovery of the Sun in 450 BC, has finally come to an end. Since scientists can never be content, they have turned to the next best thing, and decided to find planets instead.

A passionate group, the Planet Discovery Aficionados (PDA), has already started working on it! To find new planets, they have built the Large Astronomical Telescope of Amazing Might (LATAM) – a telescope that is extremely sensitive to light and thus can see even the tiniest planets.

When pointed to the sky, LATAM's field of view encompasses the portion of space enclosed strictly by a circular cone, with the vertex located at the Earth and extending infinitely away from it. The angle of the cone corresponding to the viewing area of LATAM can be adjusted by the members of PDA, and it ranges from $0°$ when the cone collapses to a line, to $90°$ when it encompasses a full half-space.

Sadly, because LATAM is so sensitive to light, if there are any stars in its field of view the picture will be a bright mess, and PDA won't be able to see any planets. They ask for your help to determine the largest possible viewing angle that does not include any stars. PDA has enough money to travel anywhere on Earth, so you are allowed to point LATAM in any direction in the sky.

Because the universe is sufficiently large, PDA considers that all the stars and the Earth are fixed 3D points, with the Earth located at $(0, 0, 0)$. The group will give you the coordinates of all the stars in the universe. They hope that with this information, you will be able to complete your task.

Input

The first line contains an integer N ($1 \leq N \leq 500$) indicating the number of stars in the universe.

Each of the next N lines describes a star with three integers X, Y and Z ($-10^3 \leq X, Y, Z \leq 10^3$), representing that the star is located at (X, Y, Z). No two stars share the same location, and no star is located at $(0, 0, 0)$.

Output

Output a single line with the largest possible angle the telescope can be set to without observing any stars. If there exists a direction where a full half-space contains no stars, report $90°$. The output must have an absolute or relative error of at most 10^{-4}.

Sample input 1	Sample output 1
6 1 0 0 -1 0 0 0 1 0 0 -1 0 0 0 1 0 0 -1	54.7356103172

Sample input 2	Sample output 2
2 314 314 314 1 1 1	90

Problem K – Keep Fighting

Bob is playing a card game where he must defeat a monster. Before the game starts, Bob's power is set to P, the monster's health is set to H, and Bob receives a deck of N cards in his hands.

Each card in the deck belongs to one of the following types:

- Multiply: a card of this type has a number X written on it. Playing it multiplies Bob's current power by X.
- Add: a card of this type also has a number Y written on it. Playing it increases Bob's current power by Y.
- Attack: a card of this type allows Bob to attack the monster. Playing it reduces the monster's current health by Bob's current power.

The game is played in turns. In each turn, Bob must play exactly one card from his hand, after which the card is moved to a discard pile. If Bob has no cards left in his hand at the end of a turn, he retrieves all cards from the discard pile and can use them again in any order.

The monster is beaten as soon as its health reaches 0 or less. Is it possible for Bob to beat the monster? If so, what is the minimum number of turns required?

Input

The first line contains three integers N ($1 \le N \le 50$), P ($0 \le P \le 10^9$) and H ($1 \le H \le 10^9$), indicating respectively the number of cards in the deck, Bob's initial power and the monster's initial health.

Each of the next N lines describes a card. The content of the line depends on the type of the card, as follows:

- Multiply: the line contains the character "*" (asterisk) and an integer X ($1 \le X \le 10^9$), representing the multiply factor provided by the card.
- Add: the line contains the character "+" (plus sign) and an integer Y ($1 \le Y \le 10^9$), indicating the increment provided by the card.
- Attack: the line contains the character "!" (exclamation mark).

Output

Output a single line with an integer indicating the minimum number of turns required to beat the monster, or the character "*" (asterisk) if it is impossible to beat the monster.

Sample input 1	Sample output 1
3 0 20 * 2 ! + 5	4

Explanation of sample 1:
To beat the monster in 4 turns, Bob can play as follows:

1. Bob plays the **+ 5** card, so his power becomes **5**.
2. Bob plays the *** 2** card, so his power becomes **10**.
3. Bob plays the **!** card, so the monster's health becomes **10**. Since now Bob has no cards in his hands, all three cards go back to him.
4. Bob plays the **!** card, so the monster's health becomes **0**, and the monster is beaten.

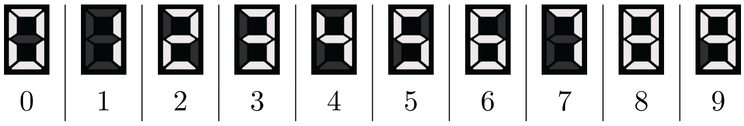
Sample input 2	Sample output 2
1 0 1 !	*

Sample input 3	Sample output 3
1 1 1 + 1	*

Problem L – LED Counter

A LED counter is a device having N positions arranged in a single row. Each position is able to display a digit from 0 to 9, and is made of seven LEDs, as this picture shows.

To display a specific digit, just the appropriate LEDs are turned on. The following picture indicates which LEDs are turned on for each possible digit.



As you can see, all but the middle LED are turned on to display the digit 0, the upper-right LED and the bottom-right LED are turned on to display the digit 1, and so on.

By turning on the appropriate LEDs in each of the N positions of the counter, the device is able to display 10^N distinct values, from $00\cdots0$ (N zeros) to $99\cdots9$ (N nines). Note that leading zeros are displayed.

Astro Void has owned a LED counter for quite some time, and so the device might have some malfunctioning LEDs. While a good LED turns on or off when needed, a malfunctioning LED is always turned on or always turned off, independently of the intended value of the counter.

Given the description of the state of each LED in Astro Void's LED counter (good LED that is turned on, good LED that is turned off, always-on LED, or always-off LED), you must tell the intended value of the counter, indicating which positions cannot be determined without ambiguity.

As an example of a LED counter with $N = 3$ positions, consider the picture below. If all the LEDs are good, then the intended value of the counter is of course 056. If the bottom LED of the first position is an always-on LED, then the intended value of the counter is still 056, because no other value would be displayed as the picture shows. However, if the bottom-left LED of the second position is an always-off LED, then the second position of the counter cannot be determined without ambiguity, since 056 and 066 would be displayed as shown.

Input

The first line contains an integer N ($1 \leq N \leq 10^5$) indicating the number of positions in the LED counter.

The i-th of the next N lines contains a string S_i of length 7 describing the seven LEDs in the i-th position of the counter. Each character of S_i describes a particular LED, from left to right and from top to bottom, that is, in the following order: upper-left, bottom-left, top, middle, bottom, upper-right and bottom-right. The character is either an uppercase letter "`G`" (good LED that is turned on), a lowercase letter "`g`" (good LED that is turned off), a plus sign "`+`" (always-on LED) or a hyphen "`-`" (always-off LED). It is guaranteed that S_i describes valid states for the LEDs in the i-th position of the counter. For instance, S_i is not "`ggggggg`", because when all LEDs are good, no digit would be displayed with all of them turned off.

Output

Output a single line with a string of N digits indicating the intended value of the counter. If a position of the counter cannot be determined without ambiguity, output the character "*" (asterisk) instead of the corresponding digit.

Sample input 1	Sample output 1
10 GGGgGGG gggggGG gGGGGGg ggGGGGG GggGgGG GgGGGgG GGGGGgG ggGggGG GGGGGGG GgGGGGG	0123456789

Sample input 2	Sample output 2
3 GGGg+GG GgGGGgG GGGGGgG	056

Sample input 3	Sample output 3
3 GGGg+GG G-GGGgG GGGGGgG	0*6

Sample input 4	Sample output 4
2 +++gG-- ---gG++	00

Sample input 5	Sample output 5
1 -+-+-+-	*

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