regionals | latin america region
icpc international collegiate programming contest
icpc.foundation
JETBRAINS
icpc global sponsor programming tools
HUAWEI
icpc diamond multi-regional sponsor

The 2026-2027 ICPC Latin America Regionals

November 7, 2026

Contest Session

This problem set contains 13 problems; pages are numbered from 1 to 21.

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

Antigua y Barbuda, Argentina, Bolivia, Brasil, Chile, Colombia, Costa Rica, Cuba, Ecuador, El Salvador, Guatemala, México, Nicaragua, Perú, Puerto Rico, República Dominicana, Trinidad y Tobago, Uruguay, and Venezuela

v1.0

General information

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

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 – Antonia's Ants

This is an interactive problem.

Antonia is studying the behavior of ants from a nearby colony. For her experiment, she will place N candies in a straight line on the ground, numbered from 1 to N in order. She can choose the color of each candy.

From previous observations, Antonia knows that each ant leaves the colony and starts at any one of the candies. It then walks along the line, repeatedly moving from its current candy to the candy immediately to its left or right. Before returning to the colony, the ant visits every candy at least once. It may visit a candy more than once and may continue walking after it has visited every candy.

To study the routes the ants take, Antonia uses a system that records only the colors of the candies they visit, in order. She needs your help to reconstruct the sequences of visited candies from these records.

Your task is to develop a program that first chooses the colors of the candies and then reconstructs the sequences of visited candies from the recorded color sequences. For this to be possible, the coloring that you choose must ensure that no two different sequences of visited candies produce the same color sequence. We call such a coloring *valid*.

A valid coloring can always be obtained by using N different colors, one for each candy. However, Antonia wants to use candies with as few different colors as possible.

Interaction

First, read a line with an integer N ($2 \leq N \leq 10^5$) indicating the number of candies.

Then output a line with an integer K followed by N integers $C_1, C_2, \ldots, C_N$ ($1 \leq C_i \leq K$), where K is the number of different colors used and C_i is the color of candy i. You must provide a valid coloring that uses the minimum number of different colors. If several such colorings exist, you may output any of them.

After that, read a line with an integer Q ($1 \leq Q \leq 100$) indicating the number of recorded color sequences. Each sequence corresponds to a different ant, and all sequences use the coloring you provided. Then, repeat the following Q times, once per color sequence:

- Read a line with an integer M ($M \geq N$) followed by M integers $S_1, S_2, \ldots, S_M$ ($1 \leq S_i \leq K$), where M is the length of the sequence of visited candies, and S_i is the color of the i-th candy in this sequence.
- Output a line with M integers $P_1, P_2, \ldots, P_M$ ($1 \leq P_i \leq N$), where P_i is the number of the i-th candy in the sequence of visited candies.

The sum of M over all recorded color sequences does not exceed 4×10^5.

Each color sequence is guaranteed to correspond to an ant that visits every candy at least once, moving only between neighboring candies.

The judge program is adaptive, which means that Q and the recorded color sequences may depend on your outputs.

Your solution will be rejected if the coloring is not valid, K is not minimum, or any reconstruction is wrong. In that case the interaction may end at any time, not necessarily immediately after the incorrect output.

Sample interaction 1

Read	Write
`2`	
	`2 1 2`
`2` `2 1 2`	
	`1 2`
`5 2 1 2 1 2`	
	`2 1 2 1 2`

Explanation of sample 1:

In this case there are $N = 2$ candies.

The number of different colors ($K = 2$) is minimum: with a single color, the sequences of visited candies 1 2 and 2 1 would produce the same color sequence. In the sample, candy 1 has color 1 and candy 2 has color 2.

Then there are $Q = 2$ recorded color sequences, each corresponding to a different ant. Taking into account the chosen colors, the color sequence 1 2 can only come from the sequence of visited candies 1 2, and the color sequence 2 1 2 1 2 can only come from the sequence of visited candies 2 1 2 1 2.

Sample interaction 2

Read	Write
`4`	
	`4 1 2 3 4`
`1` `6 2 3 4 3 2 1`	
	`2 3 4 3 2 1`

Explanation of sample 2:

Notice that this interaction would be rejected, because it uses more colors than necessary. It is included only to illustrate the interaction format.

In this case $N = 4$, $K = 4$, $Q = 1$ and $M = 6$. The color sequence 2 3 4 3 2 1 can only come from the sequence of visited candies 2 3 4 3 2 1. The coloring is valid and the reconstruction is correct. However, a valid coloring with fewer than four colors exists for $N = 4$, so K is not minimum.

Problem B – Between Two Cameras

In a rectangular room there are N sleeping cats. Each cat has a position on the floor and a weight. They do not move.

Two cameras are located at adjacent corners of the room. Each camera starts pointing toward the other camera and turns 90° toward the interior of the room. As it turns, it records each cat when it points directly at that cat. Each camera then creates a panoramic image showing the cats in the order it recorded them.

A control system compares the two panoramas. A pair of distinct cats $\{A, B\}$ is flagged if A appears before B in one panorama and after B in the other. Each flagged pair needs manual review, at a cost equal to the product of the weights of those two cats.

Given the positions and weights of the cats, you must find the sum of the costs of all flagged pairs.

Input

The first line contains two integers N ($2 \leq N \leq 2 \times 10^5$) and D ($2 \leq D \leq 10^9$), indicating respectively the number of cats and the distance between the two cameras.

The floor is represented in the plane, with the cameras located at $(0, 0)$ and $(D, 0)$. The interior of the room is on the side where the second coordinate is positive.

Each of the next N lines describes a cat with three integers X ($1 \leq X \leq D-1$), Y ($1 \leq Y \leq 10^9$) and W ($1 \leq W \leq 10^4$), where (X, Y) is the position of the cat and W is its weight.

It is guaranteed that neither camera ever points directly at two cats at once.

Output

Output a single line with an integer indicating the sum of the costs of all flagged pairs.

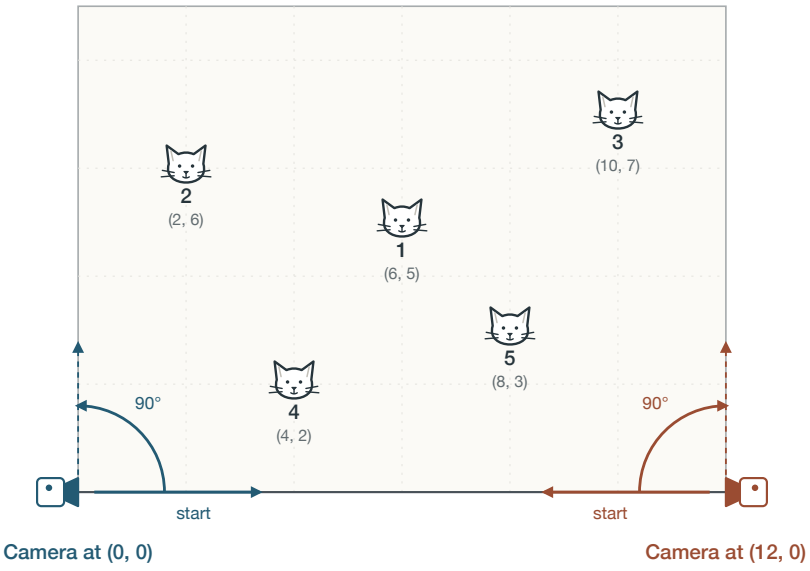
Sample input 1	**Sample output 1**
3 10 5 2 3 5 3 4 5 1 2	0

Sample input 2	**Sample output 2**
3 10 5 3 2 8 3 3 2 3 1	11

Sample input 3	**Sample output 3**
5 12 6 5 5 2 6 2 10 7 1 4 2 3 8 3 4	37

Explanation of sample 3:

For the configuration shown in the figure below, the flagged pairs are $\{5, 4\}$, $\{5, 2\}$, $\{3, 1\}$, $\{3, 2\}$ and $\{1, 2\}$, whose costs sum to $4 \cdot 3 + 4 \cdot 2 + 1 \cdot 5 + 1 \cdot 2 + 5 \cdot 2 = 37$.



Cats are numbered in input order.

RECORDING ORDER

Camera at (0, 0)	5 → 4 → 3 → 1 → 2
Camera at (12, 0)	4 → 2 → 5 → 1 → 3

Problem C – Clumsy Courier

A courier must deliver a package across a city. The city's streets form a rectangular grid with R horizontal streets and C vertical streets. On the city map, the horizontal streets are numbered from 1 to R from top to bottom, and the vertical streets from 1 to C from left to right. Intersection (r, c) is where horizontal street r meets vertical street c. The courier starts at $(1, 1)$ and must deliver the package at (R, C).

In one move, the courier can follow a street from (r, c) to an adjacent intersection:

- Down, to $(r + 1, c)$;
- Up, to $(r - 1, c)$;
- Right, to $(r, c + 1)$; or
- Left, to $(r, c - 1)$.

The courier may only travel between intersections in this grid.

The courier is in no hurry to finish the delivery. To make the trip last longer, they want to choose a route that includes exactly U Up moves and exactly L Left moves. The courier is willing to visit any intersection more than once, including the starting and ending intersections.

Count the number of different routes from $(1, 1)$ to (R, C) that stay within the grid and contain exactly U Up moves and exactly L Left moves. Two routes are considered different if their sequences of moves are different.

Input

The input consists of a single line that contains four integers R, C, U and L ($2 \leq R, C \leq 2000$ and $0 \leq U, L \leq 2000$), indicating respectively the number of horizontal streets, the number of vertical streets, the required number of Up moves, and the required number of Left moves.

Output

Output a single line with an integer indicating the number of valid routes. Since this number can be large, output the remainder of dividing it by 998 244 353.

Sample input 1	**Sample output 1**
2 2 1 0	4

Explanation of sample 1:

Each route uses two Down moves, one Up move, and one Right move. Writing directions by their first letters, the four routes are DUDR, DURD, DRUD, and RDUD. The first two routes visit the starting intersection twice, while the last two routes visit the destination twice.

Sample input 2	**Sample output 2**
3 3 1 1	280

Sample input 3	**Sample output 3**
2 163 2 0	20719340

Problem D – Divine Lights

In an old temple, there are N lamps, numbered from 1 to N. Each lamp is initially either on or off.

A priest performs a ceremony lasting N days. On day i of the ceremony, the priest toggles every lamp except lamp i: lamps that are on are turned off, and lamps that are off are turned on. The lamps remain in their resulting states until the next day.

Given the initial state of each lamp, determine the minimum and maximum numbers of lamps that are on at the end of a day during the ceremony. The initial states of the lamps must not be considered.

Input

The input consists of a single line that contains a string of N characters ($1 \leq N \leq 2 \times 10^5$). The i-th character is "`1`" if lamp i is initially on, and "`0`" if it is initially off.

Output

Output a single line with two integers representing the minimum and maximum numbers of lamps that are on at the end of a day during the ceremony, in that order.

Sample input 1	**Sample output 1**
`1011`	`1 3`

Explanation of sample 1:

In this sample $N = 4$ and the initial states of the lamps are given by "`1011`". At the end of each day of the ceremony their states are:

- Day 1: "`1100`" (2 lamps on)
- Day 2: "`0111`" (3 lamps on)
- Day 3: "`1010`" (2 lamps on)
- Day 4: "`0100`" (1 lamp on)

Thus, among the N days considered, the minimum number of lamps that are on is 1, and the maximum is 3.

Sample input 2	**Sample output 2**
`11`	`0 1`

Explanation of sample 2:

Here $N = 2$ and the initial states of the lamps are given by "`11`". At the end of day 1 the states of the lamps are "`10`", and at the end of day 2 they are "`00`". The minimum is 0 and the maximum is 1. Note that initially two lamps are on, but the initial states must not be considered.

Problem E – Escape Plan

Mrs. Worm owns a coal mine that can be represented as a network of N nodes numbered from 1 to N. Node 1 is the entrance to the mine, and for each other node i there is a one-way tunnel going from a node P_i to i, with $P_i < i$. Nodes that don't have any tunnels coming out of them are called *mining shafts*.

The Nlogonia government conducted a safety inspection of the mine and noted that, in case of an emergency, workers could be stuck inside the mining shafts, unable to escape. Mrs. Worm will certainly get sued unless she acts quickly!

To avoid this problem, Mrs. Worm decided to dig a set of one-way *escape tunnels* leading out from the mining shafts. This set must meet the following conditions:

a. Each mining shaft s has exactly one escape tunnel going from s to a node $e(s)$, in such a way that there is a path from $e(s)$ to s using only the original $N-1$ tunnels.

b. Each node has at most one escape tunnel that leads to it.

c. There is a path from each mining shaft to the entrance, if both the original tunnels and the escape tunnels can be used.

Mrs. Worm needs your help! Find a valid set of escape tunnels that saves her from being sued or tell her that it is impossible to do so.

Input

The first line contains an integer N ($2 \leq N \leq 5 \times 10^5$) representing the number of nodes in the coal mine. Each node is identified by a distinct integer from 1 to N.

The second line contains $N-1$ integers $P_2, P_3, \ldots, P_N$ ($1 \leq P_i < i$), representing that there is a tunnel going from node P_i to node i.

Output

Output a single line with N integers $E_1, E_2, \ldots, E_N$, such that $E_i = e(i)$ if node i is a mining shaft, and $E_i = 0$ otherwise. If there are multiple solutions, output any of them.

If there is no valid set of escape tunnels, output a single line with the character "*" (asterisk) instead.

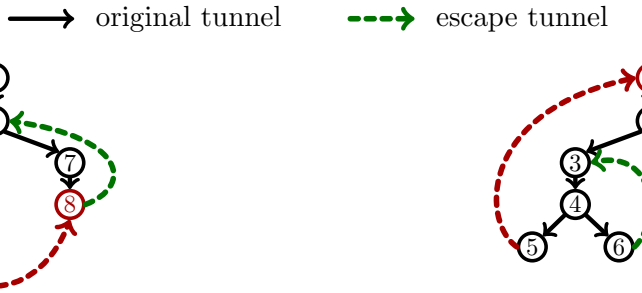
Sample input 1	**Sample output 1**
2 1	0 1

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

Sample input 3	**Sample output 3**
8 1 2 3 4 4 2 7	0 0 0 0 1 3 0 2

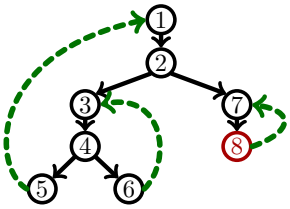
Explanation of sample 3:

Nodes 5, 6 and 8 are the mining shafts. The figure below shows four possible sets of escape tunnels for this test case. The first three sets are invalid, each one breaking one of the conditions. The last set is the proposed output $e(5) = 1$, $e(6) = 3$ and $e(8) = 2$, which is valid.

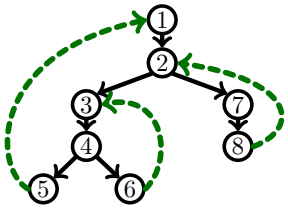


(a) Invalid: $e(6) = 8$, but there is no path from node 8 to node 6 using only the original tunnels.

(b) Invalid: two escape tunnels lead to node 1.



(c) Invalid: mining shaft 8 cannot reach the entrance.



(d) Valid: the set of escape tunnels of the output.

Problem F – Father's Footsteps

Claudio works hard and enjoys helping others. Over the years, he has learned many skills: he can work as a farmhand, carpenter, engineer, mechanic, gravedigger, businessman, or neighborhood bar owner. Whatever the job, he does it exceptionally well, even without formal training.

Today he has N jobs to do, in a fixed order, and each job requires a specific skill. Claudio needs to change tools whenever two consecutive jobs in his schedule require different skills. Notice that preparing the tools for his first job does not count as a tool change.

Claudio's son is following in his footsteps and wants to practice one of the skills. The son can choose any skill and perform all jobs requiring that skill. These jobs are removed from Claudio's schedule, while the order of all remaining jobs stays the same.

When choosing a skill to practice for the day, Claudio's son wants to help his father as much as possible, minimizing the number of tool changes that Claudio will need for the remaining jobs. What is this minimum number?

Input

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

The second line contains N integers $A_1, A_2, \ldots, A_N$ ($1 \leq A_i \leq N$), where A_i is the skill required for the i-th job.

Output

Output a single line with an integer representing the minimum possible number of tool changes.

Sample input 1	**Sample output 1**
8 1 2 1 2 2 3 2 1	2

Explanation of sample 1:

If Claudio's son performs all jobs requiring skill 1, the remaining jobs require the skills 2 2 2 3 2, and then Claudio needs to change tools twice. Choosing another skill does not reduce the number of tool changes.

Sample input 2	**Sample output 2**
2 1 2	0

Explanation of sample 2:

Choosing either skill 1 or skill 2 leaves a single job, so Claudio makes no tool changes.

Sample input 3	**Sample output 3**
5 3 3 3 3 3	0

Explanation of sample 3:

Claudio's son can choose skill 3 and perform all five jobs, leaving Claudio with no jobs and hence no tool changes.

Problem G – Gratitude Loop

The contest is over, but organizers and contestants are still thanking each other:

"Thank you!"
"No, thank you!"
"No, no, thank you!"
"No, no, no, thank you!"

This could go on forever! To save everyone some typing, they created a messaging app with just one button: Thanks!. After some effort, we were able to obtain a screenshot of the app installed on Paulo's device. We include it below for your convenience.



Of course, Paulo is not the only person using this wonderful app. There are N people using it, numbered from 1 to N. Initially, nobody has received any messages. Whenever person s presses Thanks!, that person sends exactly one message to a recipient r. The app chooses r as follows:

- If the sender s has already received any messages, then the recipient r is the person who most recently sent a message to s.
- If the sender s has not yet received any messages, then the recipient r is $s + 1$, unless $s = N$, in which case r is 1.

The message is delivered immediately, before the next button press, and the sender s becomes the person who most recently sent a message to the recipient r.

Receiving a message does not send an automatic reply. Sending a message does not delete or change any of the sender's received messages. A person can press the button repeatedly, even without receiving new messages between their button presses.

Given the sequence of button presses, determine how many messages each person receives. Count each message separately, including repeated messages from the same person.

Input

The first line contains two integers N ($2 \leq N \leq 1000$) and M ($1 \leq M \leq 1000$), indicating respectively the number of people and the number of button presses. Each person is identified by a distinct integer from 1 to N.

The second line contains M integers $A_1, A_2, \ldots, A_M$ ($1 \leq A_i \leq N$), where A_i is the person who makes the i-th button press. The presses are made in the given order.

Output

Output a single line with N integers $C_1, C_2, \ldots, C_N$, such that C_i is the number of messages received by person i during the M button presses.

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

Explanation of sample 1:

In this sample there are $N = 4$ people who make $M = 4$ button presses. The messages are sent from 1 to 2 (no messages received yet by 1), from 2 to 1 (last message received by 2 is from 1), from 3 to 4 (no messages received yet by 3), and from 2 to 1 (last message received by 2 is still from 1).

Sample input 2	Sample output 2
2 3 2 2 2	3 0

Explanation of sample 2:

Here $N = 2$ and $M = 3$. Person 2 never receives a message, so each of their three button presses sends a message to person 1.

Sample input 3	Sample output 3
3 4 1 2 3 1	2 1 1

Explanation of sample 3:

Now $N = 3$ and $M = 4$. The messages are sent from 1 to 2, from 2 to 1, from 3 to 1, and from 1 to 3.

Problem H – Hazel's Harvest

Hazel is starting a tree plantation. She likes to plan ahead and has prepared a planting schedule for N trees.

The trees will be grown elsewhere before being moved to Hazel's plantation. Each tree will be planted there in a specific year Y, when it will already be H meters tall. Once planted, it remains in the plantation and grows exactly one meter per year. Thus, in any year $t \geq Y$, the tree is $H + t - Y$ meters tall.

To plan her future sales, Hazel considers Q potential purchase orders for trees. Each order requests K trees, each at least V meters tall. An order can be fulfilled in a given year if the plantation has at least K planted trees that are each at least V meters tall.

The orders are independent: no trees are actually removed from the plantation, and each order is considered as if it were the only one. For each order, determine the earliest year in which it could be fulfilled.

Input

The first line contains an integer N ($1 \leq N \leq 2 \times 10^5$) representing the number of trees in the planting schedule.

Each of the next N lines describes a tree with two integers Y and H ($1 \leq Y, H \leq 10^9$), indicating respectively its planting year and its height in meters when planted.

The next line contains an integer Q ($1 \leq Q \leq 2 \times 10^5$) representing the number of purchase orders.

Each of the next Q lines describes a purchase order with two integers K ($1 \leq K \leq N$) and V ($1 \leq V \leq 10^{18}$), indicating respectively the number of trees requested and their minimum height in meters.

Output

For each purchase order in the input, output a line with an integer indicating the earliest year in which it could be fulfilled. It can be shown that an answer exists for each purchase order.

Sample input 1	**Sample output 1**
3 1 5 6 10 3 2 3 2 4 1 11 3 1	5 7 6

Explanation of sample 1:

For the first order ($K = 2$, $V = 4$), the first tree is already tall enough when it is planted in year 1. The third tree reaches a height of 4 meters in year 5, while the second tree has not yet been planted. Therefore, the earliest year in which the order could be fulfilled is 5.

For the second order ($K = 1$, $V = 11$), the first two trees reach a height of 11 meters in year 7, while the third tree reaches that height in year 12. Therefore, the answer is 7.

For the third order ($K = 3$, $V = 1$), every tree is tall enough from the year it is planted. However, the second tree is not in the plantation before year 6. Therefore, the answer is 6.

Sample input 2	Sample output 2
2 5 8 5 8 3 2 8 1 10 1 8	5 7 5

Explanation of sample 2:

Both trees are planted in year 5, when they are already 8 meters tall. They are separate trees, so the first order ($K = 2$, $V = 8$) could be fulfilled in year 5, but not earlier.

Both trees reach a height of 10 meters in year 7, the answer to the second order ($K = 1$, $V = 10$).

The third order ($K = 1$, $V = 8$) could be fulfilled in year 5: orders are considered independently, so neither of the previous orders removes any trees.

Problem I – Imprecise Location

A reverse geocoding service maps coordinates to geographic regions. In this problem, you must implement such a service.

You are given N geographic regions. Each region is an axis-aligned rectangle in the plane: its sides are parallel to the coordinate axes. It is described by two opposite corners (X_1, Y_1) and (X_2, Y_2), with $X_1 < X_2$ and $Y_1 < Y_2$. For any two regions, either one is strictly contained in the other or they are disjoint. Their boundaries do not touch or cross.

The service must answer Q location queries. Each query is given by three integers X, Y and R. The point (X, Y) is the measured location, and R describes the uncertainty of the measurement: the actual location can be any point inside or on the boundary of the axis-aligned query square with opposite corners $(X - R, Y - R)$ and $(X + R, Y + R)$. When $R = 0$, the query square consists of the single point (X, Y).

It is guaranteed that no corner of a query square lies on the boundary of any region. However, the boundary of a query square may intersect region boundaries.

For each query, determine the region of smallest area that is guaranteed to contain the actual location. In other words, find the region of smallest area that contains the entire query square, or report that no such region exists.

Input

The first line contains an integer N ($1 \leq N \leq 2 \times 10^5$) indicating the number of geographic regions. Each region is identified by a distinct integer from 1 to N.

The i-th of the next N lines describes region i with four integers X_1, Y_1, X_2 and Y_2 ($-10^9 \leq X_1, Y_1, X_2, Y_2 \leq 10^9$, $X_1 < X_2$ and $Y_1 < Y_2$), where (X_1, Y_1) and (X_2, Y_2) are two opposite corners of the corresponding rectangle.

The next line contains an integer Q ($1 \leq Q \leq 2 \times 10^5$) indicating the number of location queries.

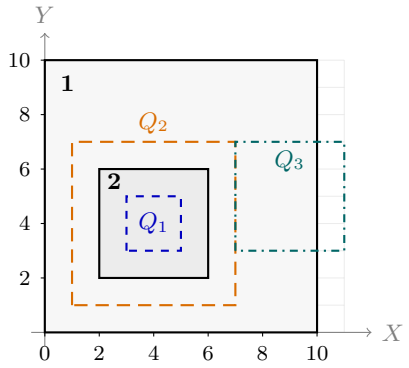
Each of the next Q lines describes a query with three integers X, Y and R ($-10^9 \leq X, Y \leq 10^9$ and $0 \leq R \leq 10^9$), representing the query square with opposite corners $(X - R, Y - R)$ and $(X + R, Y + R)$. The corners of this square may lie outside the coordinate range given for X and Y.

Output

For each query in the input, output a line with an integer identifying the region of smallest area that contains the entire query square. If no region contains the entire query square, output a line with the character "*" (asterisk) instead.

Sample input 1	**Sample output 1**
2 0 0 10 10 2 2 6 6 3 4 4 1 4 4 3 9 5 2	2 1 *

Explanation of sample 1:

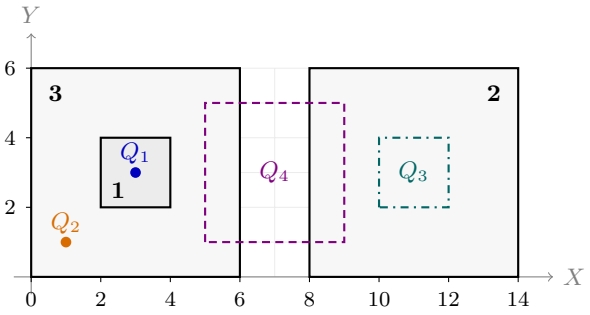


Solid rectangles: regions
Q_i: i-th query (square or point)

$Q_1 \to 2$
$Q_2 \to 1$
$Q_3 \to *$

Sample input 2	Sample output 2
3 2 2 4 4 8 0 14 6 0 0 6 6 4 3 3 0 1 1 0 11 3 1 7 3 2	1 3 2 *

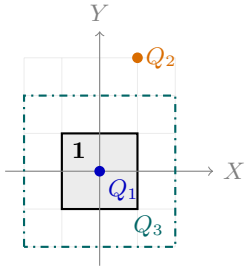
Explanation of sample 2:



$Q_1 \to 1$
$Q_2 \to 3$
$Q_3 \to 2$
$Q_4 \to *$

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

Explanation of sample 3:



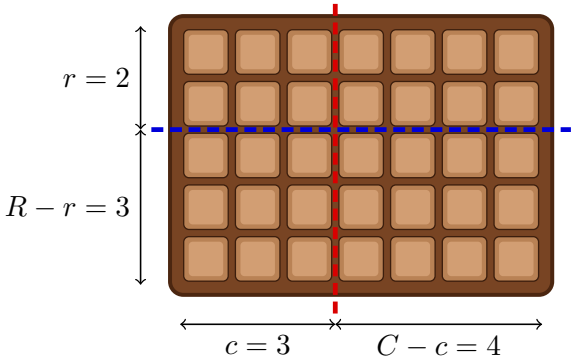
$Q_1 \to 1$
$Q_2 \to *$
$Q_3 \to *$

Problem J – Just Four Pieces

Grandma bought a chocolate bar for her four grandchildren. The bar is made up of small chocolate squares arranged as a grid of R rows and C columns. The grandchildren are all of different ages, and Grandma's rule is simple: the older the child, the more chocolate they get. To share the entire bar without waste, she wants to split it into four rectangular pieces with pairwise distinct numbers of squares.

Grandma knows that snapping a chocolate bar by hand can be irregular. To ensure clean cuts, after placing the bar flat on a cutting board she will make two straight cuts with a knife along the grooves between squares: one horizontal cut between rows r and $r+1$ ($1 \leq r < R$), and one vertical cut between columns c and $c+1$ ($1 \leq c < C$). Each cut crosses the entire bar.

In the picture below, a chocolate bar with $R = 5$ and $C = 7$ is cut by choosing $r = 2$ and $c = 3$, which yields pieces with 6, 8, 9 and 12 squares.



Help Grandma choose where to cut: find a pair of integers r and c such that the four resulting pieces have pairwise distinct numbers of squares, or determine that no such pair exists.

Input

The input consists of a single line that contains two integers R and C ($2 \leq R, C \leq 2 \times 10^9$), indicating respectively the number of rows and columns of the chocolate bar.

Output

Output a single line with two integers r and c, indicating where the chocolate bar can be cut to obtain four pieces with pairwise distinct numbers of squares. If there are multiple solutions, output any of them.

If four pieces with distinct numbers of squares cannot be obtained, output a single line with the character "*" (asterisk) instead.

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

Sample input 2	**Sample output 2**
5 7	2 4

Sample input 3	**Sample output 3**
2 2	*

Problem K – Kayaking

Your friend Luis is an expert kayaker planning to navigate a treacherous river. The river is modeled as a $2 \times N$ grid, completely enclosed by border cells to form a grid with 4 rows and $N + 2$ columns.

Border cells describe the surroundings of the river. Each of them is either a safe zone, like a calm eddy or the shore, represented by "`.`" (dot), or an obstacle, like a rock or a waterfall, represented by "`#`" (hash).

In each river cell, the current flows left, right, up or down, represented respectively by "`<`" (less-than sign), "`>`" (greater-than sign), "`^`" (caret) or "`v`" (lowercase letter v). Some river cells may have an unknown flow direction, represented by "`?`" (question mark).

A river cell is *safe* if following the current from that cell eventually leads to a safe zone "`.`". Otherwise, the river cell is *unsafe*: the current hits an obstacle "`#`" or enters a loop within the river.

For example, consider the first sample input, where $N = 2$. The top-left river cell is safe since the current flows into the safe zone on its left. The other three river cells are unsafe: the current from the bottom-left river cell leads to the obstacle directly below it, while the two river cells on the right form a loop.

A *configuration* of the river is obtained by assigning any of the four directions "`<`", "`>`", "`^`" or "`v`" to each unknown flow direction "`?`"; each unknown flow direction can be assigned independently of the others.

Luis needs your help to assess the river's safety. For each integer k from 0 to $2N$ inclusive, you must determine the number of configurations having exactly k safe river cells.

Input

The first line contains an integer N ($1 \leq N \leq 1000$).

Each of the next 4 lines contains a string of $N + 2$ characters representing a row of the grid, as described in the statement.

Output

Output a single line with $2N + 1$ integers $C_0, C_1, \ldots, C_{2N}$, where C_k indicates the number of configurations having exactly k safe river cells. Since these numbers can be large, output the remainder of dividing each of them by 998 244 353.

Sample input 1	**Sample output 1**
2 ##.# .<v. .v^. ##.#	0 1 0 0 0

Sample input 2	**Sample output 2**
1 #.# .?. .?. #.#	1 0 15

Sample input 3	Sample output 3
2 ##.# .??. .^<# ...#	6 4 0 0 6

Sample input 4	Sample output 4
4 ..#..# #>v^?. #?<??. #.#.#.	0 18 48 39 87 3 12 9 40

Sample input 5	Sample output 5
8 .########. #????????# #???^????# .########.	75497471 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Problem L – Lookout Towers

A kingdom has N villages connected by $N-1$ two-way roads, so that it is possible to travel between any two villages using the roads. The distance between two villages is the minimum number of roads needed to travel from one village to the other.

Some villages have a watchtower, and there are at least two watchtowers in the kingdom. The king wants to know how close the two nearest watchtowers are, and how many pairs of watchtowers are that close.

Input

The first line contains an integer N ($2 \leq N \leq 2 \times 10^5$) indicating the number of villages. Each village is identified by a distinct integer from 1 to N.

The second line contains N integers $W_1, W_2, \ldots, W_N$, where $W_i = 1$ if village i has a watchtower, and $W_i = 0$ otherwise. At least two of these values are 1.

Each of the next $N-1$ lines contains two integers U and V ($1 \leq U, V \leq N$ and $U \neq V$), indicating that there is a two-way road between villages U and V. It is guaranteed that it is possible to travel between any two villages using the roads.

Output

Output a single line with two integers: the minimum distance between two different watchtowers, followed by the number of unordered pairs of watchtowers at that distance.

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

Explanation of sample 1:

In this sample we have $N = 5$ villages connected in a line, with watchtowers at both ends and in the middle (1–2–3–4–5). The minimum distance is 2, with pairs $\{1, 3\}$ and $\{3, 5\}$ at that distance.

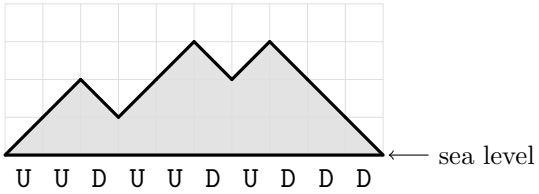
Sample input 2	**Sample output 2**
6 1 0 0 1 1 1 2 1 4 2 1 3 6 3 2 5	2 4

Explanation of sample 2:

Now we have $N = 6$ villages, with watchtowers at villages 1, 4, 5 and 6. The pairs $\{1, 4\}$, $\{1, 5\}$, $\{1, 6\}$ and $\{4, 5\}$ are all at distance 2. No other pair is at that distance or closer.

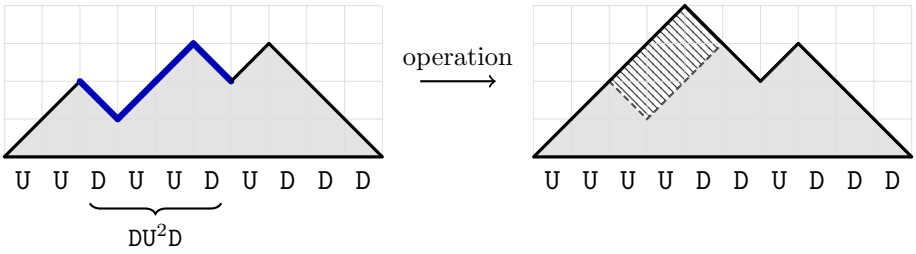
Problem M – Moving Mountains

The islands of Polylogonia and Quasilogonia have mountain ranges whose profiles are made of slopes at 45°. A profile is described from left to right by a string of "U" (uphill) and "D" (downhill) characters. The two profiles have the same length. Each mountain range starts and ends at sea level and never goes below it. Since it starts and ends at sea level, each profile has the same number of "U" and "D" characters. The figure below shows the profile "UUDUUDUDDD".



The Island Contour Planning Company (ICPC) owns a machine that is capable of filling valleys in a very specific way. In one operation, it chooses an island and covers an entire uphill slope with earth. This slope runs from the bottom of a valley to the first peak to its right. More precisely, the operation replaces a substring of that island's profile of the form "$\texttt{DU}^k\texttt{D}$" ($k \geq 1$) with "$\texttt{U}^k\texttt{DD}$", where "$\texttt{U}^k$" denotes k consecutive "U" characters. All other characters remain unchanged. The machine cannot remove earth, only add it in the way just described.

The figure below shows one operation of the machine: the part "$\texttt{DU}^2\texttt{D}$" marked by the brace on the left becomes "$\texttt{U}^2\texttt{DD}$" on the right, where the hatched strip is the earth that was added.



The ICPC wants to make the two profiles identical using its machine. It can be shown that this is always possible. Find the minimum total number of operations needed.

Input

The first line contains a string P ($2 \leq |P| \leq 2 \times 10^5$), representing the profile of Polylogonia.

The second line contains a string Q ($|Q| = |P|$), representing the profile of Quasilogonia.

Each string is made of the uppercase letters "U" and "D", indicating respectively uphill and downhill sections of the corresponding mountain range. In each string the number of "U" and "D" characters is the same, and no prefix has more "D" characters than "U" characters.

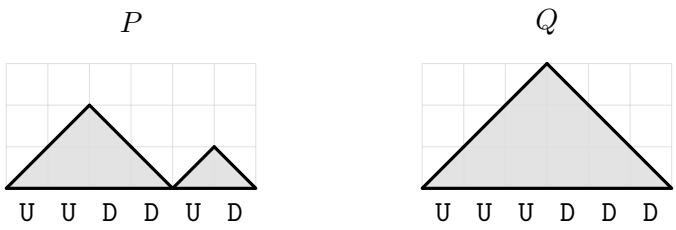
Output

Output a single line with an integer indicating the minimum total number of operations needed to transform the profiles P and Q into the same profile.

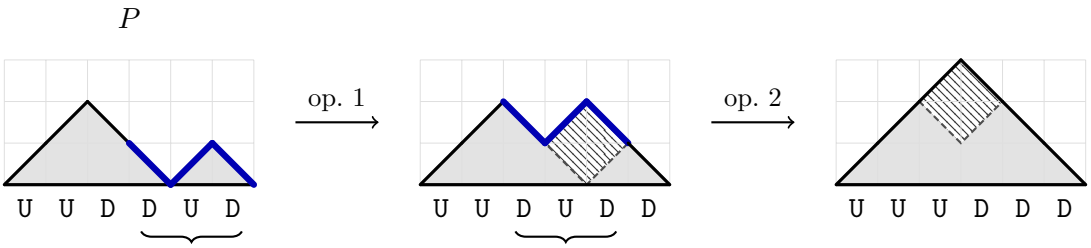
Sample input 1	Sample output 1
UUDDUD UUUDDD	2

Explanation of sample 1:

The profiles of the two islands are depicted below.



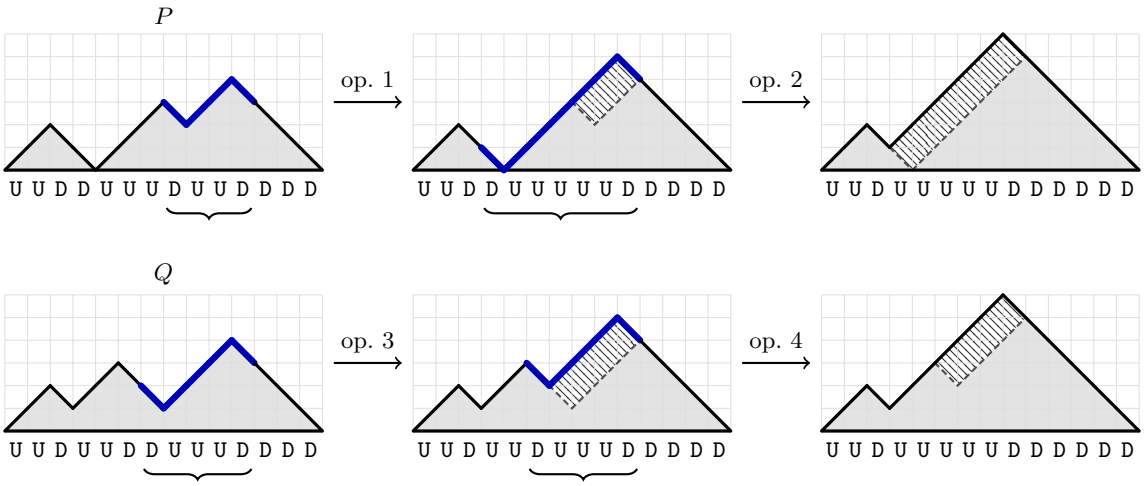
Q has no "D" followed by "U", so no operation can be applied to it, which means that both islands must end up with the profile "UUUDDD". P can be turned into this profile with two operations, as shown below.



Sample input 2	Sample output 2
UUDDUUUDUUDDDD UUDUUDDUUUDDDD	4

Explanation of sample 2:

Both P and Q can be turned into the profile UUDUUUUUDDDDDD with two operations each, for a total of four operations, as shown below.



Sample input 3	Sample output 3
UDUUDUUUDDDD UDUUUDDUUDDD	3