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

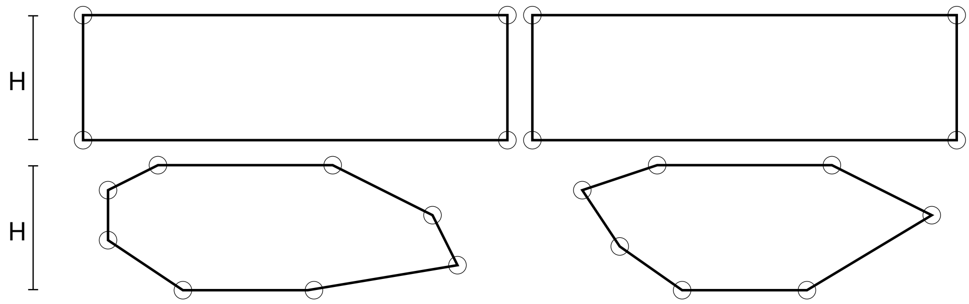
Problem A – Arranging tiles

Author: Guilherme A. Pinto, Brasil

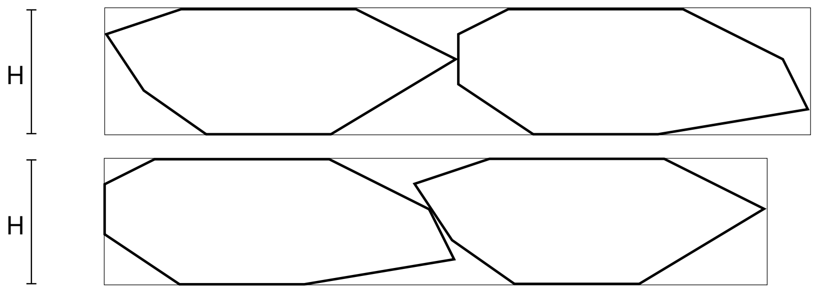
A set of rectangular stone tiles, all of them having the same height H, had their original four corners cut in different ways so that two properties were kept:

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 ($1 \le L, C, N \le 10^5$), 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 ($1 \le P, X \le C$ and $0 \le A, B \le 10^8$), 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

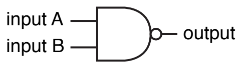
Sample input 3	Sample output 3
3 2 8 13 1 4 12 2 1 16	25

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

Problem G – Gates of uncertainty

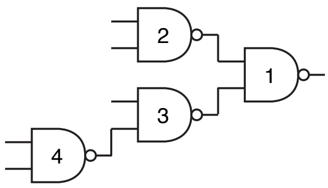
Author: Ricardo Anido, Brasil

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 ($1 \le N \le 10^5$) 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 ($0 \le X, Y \le N$ and $-1 \le F \le 1$). 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`

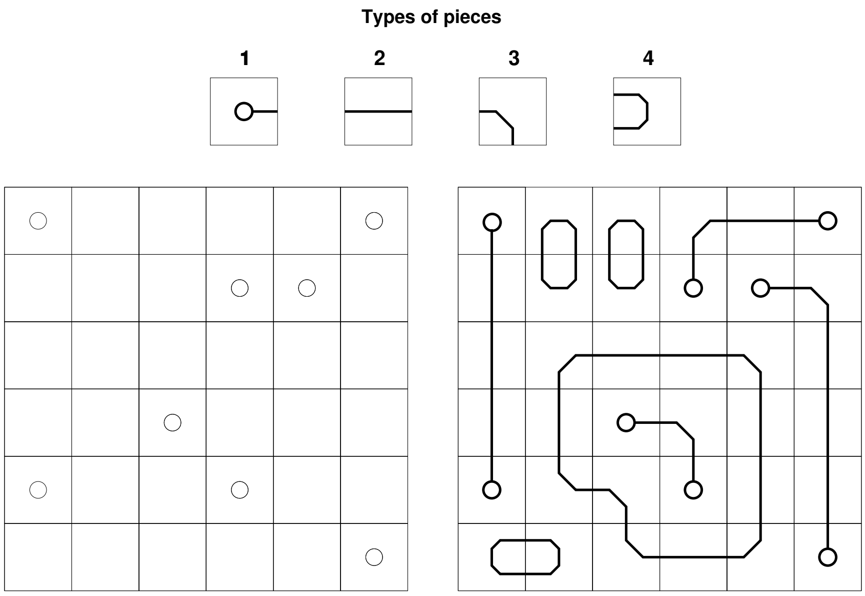
Sample input 3	Sample output 3
`PRP`	`0`

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`

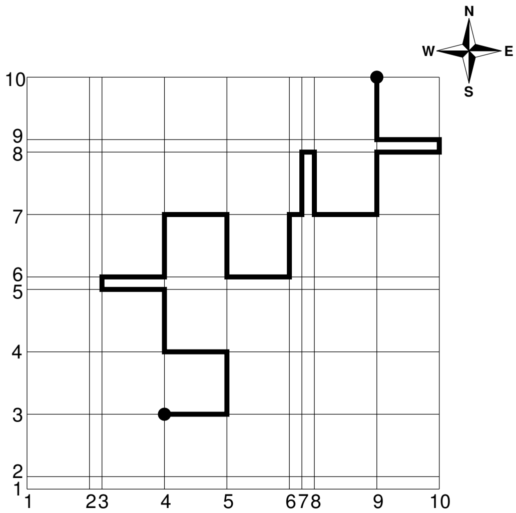
Sample input 4	Sample output 4

`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.