Problem A. Arithmetic Expression from an Integer

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

For decimal representation of positive integer N it is allowed to construct an arithmetic expression by inserting one of signs '+', '-' and '*' between any of two digits or not inserting anything. For example, from number 123 can be built expressions such as $1 + 2 - 3$, $1 * 2 + 3$, $12 + 3$ etc.

Print maximal integer K which may be obtained as value of such a expression for given N.

Input

Input consists of one positive integer N ($1 \le N \le 10^{100}$).

Output

Print one integer — answer to the problem.

Example

standard input	standard output
9	9

Problem B. Barbara’s Robot

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Barbara is finishing her summer job in the lab testing a new 3D printed robot which is being taught to manipulate small objects.

To test the robot’s abilities, a simple device called *the sandbox* is used. It is a thin transparent square array of $N \times N$ square slots. Each slot contains a token, in the shape of an ASCII character (for better recognition), which is held in place in the slot by small magnets.

The sandbox can be rotated by 90 degrees around the axis perpendicular to its surface or flipped by 180 degrees around one of its four axes parallel to the sandbox surface.

The robot simulates sandbox rotations and flips by the following process. It removes the tokens from their slots and puts them back into different slots so that the contents of the sandbox looks exactly as if the whole sandbox was rotated or flipped. If, for achieving the desired effect, it is necessary to rotate or to flip particular tokens in their new positions the robot does it as well.

The sandbox remains stationary during the whole process.

For example, if the upper left corner of the sandbox contains a token which looks like symbol ‘<’ (less than) then after flipping the sandbox around the vertical axis this token is moved to the upper right corner where it looks like the symbol ‘>’ (greater than). Then, after left rotation the same token is moved back to the upper left corner where it looks like the symbol ‘^” (caret). Barbara has programmed the robot to perform a long sequence of successive flips and rotations.

To check the correctness of the robot’s algorithms, she needs to know in advance how the sandbox should look when the robot finishes its work.

Input

We suppose that a token on the sandbox can be shaped as any of the following so called symmetric characters: ‘<’, ‘>, ‘^”, ‘v’, ‘o’, ‘x’, ‘|’, ‘-’, ‘/’, ‘\’.

When a symmetric character is rotated or flipped it either remains the same or it becomes another symmetric character whose shape is the most similar to the rotated/flipped one.

There are 15 or less test cases. Each case starts with a line containing one integer N ($1 \le N \le 100$). Next, there are N lines representing the initial state of the sandbox.

Each line contains a string which consists of exactly N symmetric characters. Each character represents one slot on the sandbox and the order of symbols in the input corresponds to the order of the tokens on the sandbox. No slot on the sandbox is empty. After N lines, there is a line with a command string specifying the flips and rotations that has to be carried out. A command string consists of command characters, each command character specifies one rotation or flip as follows: ‘<’ (rotation left), ‘>’ (rotation right), ‘-’ (flip around horizontal axis), ‘|’ (flip around vertical axis), ‘
” (flip around main diagonal), ‘/’ (flip around anti-diagonal). Two successive command characters are separated by single space. The robot has to respect the order of commands in the command string. The number of commands is always positive and at most 10^6 .The decimal ASCII codes of the characters relevant to this problem are 45 (‘-’), 47 (‘/’), 60 (‘<’), 62 (‘>’), 92 (‘\’), 94 (‘^’), 111 (‘o’), 118 (‘v’), 120 (‘x’), 124 (‘|’).

It is guaranteed that sum of all N in one input does not exceed 500.

Output

For each test case, print N lines specifying the final position and orientation of the tokens on the sandbox. The output format of the sandbox representation is identical to the input format except for the size of the sandbox which should not be printed.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td>3
o^-
/v|
vx^
< |
5
x>-o\
voooo
|ooo/
ooo/v
\o/vv
| \ |</td><td>>-|
x<>
</o
<</o\
</ooo
/ooo|
oooo^
\o-<x</td></tr>
</table>

Problem C. Colors on a Stadium

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

You are visiting your friends on a summer holiday sports camp. When you arrive there is a volleyball tournament going on. Various clubs from the villages in the camp neighborhood are taking part in the tournament. Each club is recognized by a color which is different from the colors of all other clubs and also the fans of each particular club are wearing t-shirts with the color of their club.

Amazingly, the stadium on which the fans are sitting is suited very well for the tournament: The number of seat rows and also the number of seat columns (perpendicular to the seat rows) is equal to the number of the clubs participating in the tournament. Moreover, quite unbelievably, the number of fans of each club is also equal to the number of clubs.

The last game of the tournament is to begin shortly. To make the display on the stadium more attractive, the fans decided to arrange themselves in such way that each fan will occupy exactly one seat and each row and each column will be occupied by fans of all clubs. After some scramble, they manage to achieve this configuration and the game is about to start. Suddenly, someone in the crowd around you points to the stadium and shouts "Hey, John is not wearing his club's color!". You do not know who John is and therefore you do not know where is he sitting or what color he should be wearing. Nevertheless, if John is the only person on the stadium who failed to be dressed in proper color then it is possible, just by careful observation of the stadium, to find him and determine the color of his club.

Input

In this problem, we denote the colors of the clubs by capital letters A, B, ..., Z. There are In this problem, we denote the colors of the clubs by capital letters 'A', 'B', ..., 'Z'.

There are 350 or less test cases. Each case starts with a line containing one integer N ($3 \leq N \leq 26$) which is the number of teams taking part in the tournament. Next, there are N lines describing the colors worn by the fans on the stadium. Each line corresponds to one row of sets and it contains one string of length N. Each character in the string represents the color which the corresponding fan is wearing. The order of seats on the stadium is the same as the order of characters in the input.

It is guaranteed that sum of N for all test cases in one input does not exceed 5000.

Output

For each test case, print a single line with two integers R, C, and a character V . The values of R and C specify the row and the column on the stadium where John is sitting and the letter V specifies the color of his club. Rows and columns are numbered $1, 2, \ldots, N$. The successive values on the output line should be separated by one space.

Examples

standard input	standard output
6 OEYCDK EYOKCD KDCEOY CKHOYE YOEDKC DCKYEO 3 IWL GIW WLI	4 3 D 2 1 L

Problem D. Divisibility by Six

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Given the integer sequence A_i, built in the next way: $A_1 = 1$, $A_2 = 12$, $\ldots$ $A_{10} = 12345678910$, $\ldots$, i.e. decimal representation of A_i is decimal representation of A_{i-1}, concatenated with decimal representation of i.

Given N find if A_N is divisible by 6.

Input

First line of the input contains one integer N ($1 \leq N \leq 10^{18}$).

Output

Print "`Yes`", if A_n is divisible by 6, and "`No`" otherwise

Example

standard input	standard output
1	No
2	Yes

Problem E. Epic Win

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Johnny is playing *rock, paper, scissors* with his friend Bob. The game consists of separate rounds during which both players at the same time show with their hands a symbol of either rock, paper, or scissors. The player who shows a stronger symbol wins the round; if both players show the same symbol the round is considered a tie. The following rules show the relative strength of symbols:

- scissors cut paper (scissors are stronger than paper),
- paper covers rock (paper is stronger than rock),
- rock crushes scissors (rock is stronger than scissors).

The player who wins more rounds wins the entire game.

Mastering the game is Bob's one and only goal in life. After quite a few years of training, he prepared a long list of symbols that he considers to be the best possible strategy, and memorized it. Johnny has accidentally found a printout with the sequence and of course decided to prank Bob: he wants to win the game (that is to win strictly more rounds than his friend), and he wants to do so "epically". The win is epic when Johnny changes the symbol he's showing the least number of times possible. There's not much time until the match begins and Johnny still doesn't know what is the minimal number of changes he needs to make. Help him compute it.

Input

The first and only line of input contains a string of letters of length no greater than 10^6, where letter at position i specifies the symbol that Bob will show during round i. Each letter is either `R`, `P` or `S`, which denote respectively rock, paper, and scissors.

Output

In the first and only line you should output a single integer: the minimum possible number of changes of shown symbol that Johnny needs to make in order to win the game.

Example

standard input	standard output
RPSPRP	0
RRPPSS	1

Note

In Sample 1, Johnny can win this game by one point without making any changes, he just needs to show scissors all the time. Then:

- he wins three rounds (second, fourth, and sixth), in which Bob shows paper,
- he ties in third round, in which Bob also shows scissors,
- he loses two rounds (first and fifth), in which Bob shows rock.

In Sample 2, Johnny would make no changes he can only end the game with a tie. If he makes a single change there are many ways to win this game. One of possible ways is to show paper in the first two rounds and then change symbol to scissors for the rest of the game. Then:

- showing paper he wins first and second round, in which Bob shows rock,
- showing scissors he wins third and fourth round, in which Bob shows paper,
- continuing to show scissors he ties fifth and sixth round, in which Bob also shows scissors.

Problem F. Four Cards

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Walter is spending holidays at the farm of his great-grandfather. Once he has found four old poker cards among piles of various dusty junk. The cards look quite antiquated and interesting.

He starts to lay down some of those cards one by one, side by side, on the floor when he suddenly notices that there sometimes appears to be some kind of order in their sequence. Pairs of successive cards are either of the same rank or of the same suit. “This might be a nice little puzzle,” says Walter to himself. “I wonder if I can rearrange the sequence so that each two consecutive cards share either the rank or the suit...”

Help Walter determine whether his puzzle is solvable.

Input

There are 5000 or less test cases.

Each test case consists of a single line on which all cards in the pack are listed. The list starts with one integer L ($1 \leq L \leq 4$), denoting the number of cards selected by Walter, followed by a space and L card descriptions. Each card is described by a two character string. The first character denotes the rank of the card (‘`A`’=Ace, ‘`2`-‘`9`’, ‘`X`=10, ‘`J`=Jack, ‘`Q`=Queen, ‘`K`=King) and the second character denotes the suit of the card (‘`C`=Clubs, ‘`D`’=Diamonds, ‘`H`=Hearts, ‘`S`’=Spades). The successive card descriptions are separated by one space.

Output

For each test case, print a single line with the string “`YES`” if the puzzle is solvable or a line with the string “`NO`” if the puzzle is not solvable.

Example

standard input	standard output
4 2C 2D 2H 2S	YES
4 5C 4H AS 9D	NO
1 AS	YES

Problem G. Grapes

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Hansel received a bunch of grapes from his mother. He is supposed to share it with Gretel, who likes grapes very much. She is rather predictable, so Hansel knows that as soon as he breaks one twig of the bunch, she will insist that, because she is a girl, she should be allowed to choose one of the two obtained parts. Of course she will take the one with the larger number of berries. Hence Hansel would like to choose the twig as to guarantee that the other part (which he gets to keep) has as many berries as possible. Help him to determine how many berries he can guarantee for himself.

Hansel is using a rather peculiar terminology when talking about a bunch of grapes. Any such bunch has a tree structure: it consists of *grapes* and *twigs*, and every twig directly connects two different grapes. Any two grapes are connected with a uniquely determined sequence of twigs. One grape is distinguished and called the *root*. Grapes different than the root such that there is exactly one twig connecting them to other grapes are called the *berries*, and all other grapes are called the *joints*.

Input

The first line of input contains one integer n ($2 \le n \le 1\,000\,000$) denoting the number of grapes in the bunch. Grapes are numbered from 1 to n, where the root has number 1. The next $n-1$ lines describe the twigs, each in a separate line. Each of these lines contains two integers a and b ($a \ne b$, $1 \le a, b \le n$) denoting that grapes with the corresponding numbers are directly connected with a twig.

Output

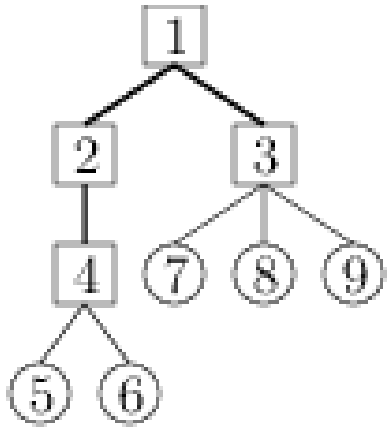
The first and only line of output should contain the maximum number of berries that Hansel can guarantee for himself by choosing the twig appropriately.

Example

standard input	standard output
9 1 2 1 3 2 4 4 5 4 6 3 7 3 8 3 9	2

Note

The following figure shows the bunch from the example. Each berry is represented by a circle, and all other grapes (that is joints) are represented by squares. Segments connecting circles and squares represent twigs. Johnny can guarantee getting berries with numbers 5 and 6 by breaking any of the twigs represented by bold segments. The remaining berries with numbers 7, 8, 9 go to Gretel.


1
2
3
4
7
8
9
5
6

Problem H. Hydrology

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Fatima is a hydrologist. She currently studies water circulation in her country river basins. She collects snapshots from various meteorological stations that measure diverse climate-related values. Fatima then searches for interesting patterns in those snapshots. She uses a program which reads incoming snapshots' data in real time and outputs those snapshots which are interesting in some way. The decision whether a snapshot is "interesting" is based on a fixed set of conditions, such as "the value is greater than the average of the last two hours" or "the value is lower than anything else in the last five minutes" which are easy to program into a computer.

Today, Fatima is in doubt about her yesterday's results and she came to see you, an experienced programmer. She thinks that her program did not evaluate the data correctly and she asks you to help her verify its results.

In particular, she brings the complete sequence of snapshots and describes the set of conditions to you. Your program has to read the snapshots and produce the output according to the conditions. Fatima will then compare the output of your program to the output of her program and decide what has to be done next.

Input

There are 150 or less test cases.

The first line of each test case contains one integer N ($1 \le N \le 10^5$), giving the number of snapshots. Then there are N lines, each describing one snapshot. The line contains two integers T_i and V_i ($1 \le T_i \le 10^9$, $1 \le V_i \le 10^4$), meaning that the snapshot value V_i was acquired in time T_i.

The times are given in seconds elapsed since some fixed moment in the past and they form a strictly increasing sequence (for all i, k such as $1 \le i < k \le N$, $T_i < T_k$).

The next line of the input contains one integer C ($1 \le C \le 10$), the number of conditions to evaluate. Each of the following C lines specifies one condition C_j. The line contains three tokens separated with a space:

1. A relation operator R_j, which is either "`gt`" (greater than) or "`lt`" (less than).
2. An aggregate function F_j, one of the "`min`" (minimum), "`max`" (maximum), or "`avg`" (average).
3. An integer number L_j specifying the length of the time interval to be concerned, in seconds.

In general, a condition applied to a snapshot value V_i checks how V_i is related to some aggregate feature of the snapshots which were acquired before V_i . The function F_j specifies exactly that feature.

To be more specific, let S_{ij} be the set of all snapshots which were acquired before V_i but no more than L_j seconds earlier. The snapshot value V_i satisfies the condition C_j if and only if the relation V_i, R_j, F_j (S_{ij}) holds. For example, the snapshot value 800 in conjunction with "`lt min 300`" can be read as "is 800 less than the minimum snapshot value acquired in the previous 5 minutes before this 800 was obtained?". Note that snapshot V_i is not an element of S_{ij}.

Output

For each condition, print one integer: the number of snapshots whose values satisfy that particular condition. If there are no snapshots in the time interval specified by the condition, the condition is never considered satisfied.

Example

standard input	standard output
10 60 30 120 28 180 35 240 34 300 40 360 31 420 28 480 2 540 42 600 30 2 gt avg 7200 lt min 300	4 2

Problem I. Industrial Buildings

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Two straight roads A and B perpendicular to each other start at the common crossing. Road A runs eastward and road B runs northward.

The roads are part of a large industrial system being built in the area and they should be connected by a tunnel. Unfortunately, the connection cannot be built directly at the intersection.

Additionally, there are some buildings standing in the corner of the field bordered by the roads. The buildings represent obstacles to the tunnel. After some negotiations and considering various technical limitations, the analysts of the tunnel building company constructed a set of critical points, determined by the obstacles. Then they suggested that the tunnel should connect the roads in a way that satisfies the following criteria:

- The tunnel should run in a straight line.
- The tunnel should not run between any critical point and the corner of the field at the roads crossing.
- The length of the tunnel should be the minimum possible.

Now, your task is to determine the length of the tunnel.

Input

There are 200 or less test cases. Each case starts with a line containing one integer N ($1 \leq N \leq 10^6$) which specifies the number of critical points. Next, there are N lines representing the points, each line describes one of them. A point P is represented by two integers a, b ($1 \leq a, b \leq 10^4$) separated by space. Integer a is the distance from P to road A and integer b is the distance from P to road B. All distances are in meters. You may suppose that the lengths of the roads and also the size of the field are not limited. All coordinate pairs (a, b) in one test case are unique.

Sum of all N in one input does not exceed $1.1 \cdot 10^6$.

Output

For each test case, print a single line with one floating point number L denoting the minimum possible length of the tunnel expressed in meters. L should be printed with the maximum allowed error of 10^{-3}.

Example

standard input	standard output
7 5 1 9 1 6 2 1 3 8 4 4 5 2 8	16.648

Problem J. Jumps

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Dave is the director of the Summer school of skydiving. Being a responsible director, he keeps a list of jumps of each trainee in the school. After each jump, Dave appends a note to the lists of jumps of each trainee participating in that particular jump. The note is very simple, it just indicates the type number of the parachute. In this way, each trainee jump history is characterized by a list of numbers.

At the end of the season, Dave wants to categorize the trainees according to their experience with different brands of parachutes.

Two trainees belong to the same category if they have used the same types of parachutes. It does not matter how many times they have jump with any particular parachute type, what does matter is the set of the parachute types they have jumped and that has to be the same.

There are exactly nine types of parachutes in Daves school, and no trainee has jumped more than nine times, so Dave expresses each trainee list as an integer consisting of digits $1, 2, \ldots, 9$ and smaller than 10^9. He thinks that this representation will help him to process the lists programmatically by a computer.

For example, the trainees characterized by integers 234423 and 342 belong to the same category, while the trainees characterized by integers 118821 and 1189821 belong to different categories. Help Dave to calculate how many different categories of trainees attended the school this season.

Input

There are 20 or less test cases. Each case starts with a line containing one integer N ($1 \le N \le 1000$) representing the number of trainees. Next, there are N lines, each line contains one integer representing the list of jumps of one particular trainee.

Output

For each test case, print a single line with one integer C specifying the number of different trainee categories in the school.

Example

standard input	standard output
5 132 42 3312 43 24424 3 222 22 2	3 1