

Welcome to
Santa's Algorithmic Challenge!

Good luck and have a great time!
Santa and the organizers

Problem A. Anagrams

Input file: stdin or input.txt
Output file: stdout or output.txt
Time limit: 1 second(s)
Memory limit: 256 MiB

The girl in the painting wrote a letter to Santa asking for a puppy, but her letter ended up in Satan's mailbox instead.

As for Santa himself, he has recently also been receiving some downright *disturbing* mail... He decided to ask the Laplandish Post to investigate a theory he came up with. To make things easier for the Post, he wants to supply them with a tool which, given two words, determines whether they are anagrams of each other. Help him write such a tool!

Input

The input consists of two lines. Each line contains a single nonempty word of at most 100 uppercase English letters.

Output

On a single line, output YES if it is possible to obtain the second word by rearranging the letters in the first word, or NO otherwise.

Examples

stdin or input.txt	stdout or output.txt
SANTA SATAN	YES
SANTASSON SONGOKU	NO
ADMIRER MARRIED	YES



The Favorite by Omar Rayyan

Problem B. Slim Chances

Input file: `stdin` or `input.txt`
Output file: `stdout` or `output.txt`
Time limit: 1 second(s)
Memory limit: 256 MiB

Santa Claus is going to visit a certain student at EPFL. However, the student has been very naughty all year, and does not really deserve a present. (No, it's not about you. Promise.) Instead of giving them old socks, coal or an empty box, Santa has decided to leave it to chance – he has selected two positive integers n and k and is going to flip n coins. The student will be getting a present only if the number of heads is exactly k . To make matters even worse, Santa's coin is biased – it lands on heads with probability $1/3$.

Santa computed the probability that the student will get a present, chuckled to himself, and flipped the first coin. It was heads. Given this knowledge, the probability that the student will get a present may have changed. To check his calculations, Santa is asking you: did it **INCREASE**, **DECREASE**, or remain the **SAME**?

Input

In this problem, every input file contains multiple testcases. The first line of the input contains the number of testcases T ($1 \leq T \leq 1000$). Next, T testcases follow, each on a separate line. Every testcase is a space-separated pair of numbers n, k with $1 \leq n, k \leq 10^6$.

Output

For every testcase, output the corresponding answer on a single line: **INCREASE**, **DECREASE**, or **SAME**.

Examples

<code>stdin</code> or <code>input.txt</code>	<code>stdout</code> or <code>output.txt</code>
2	INCREASE
2 1	DECREASE
5 1	

Note

There is no guarantee that the student had any chances of receiving a present in the first place. Some people are just *really* naughty.

Problem C. Operators

Input file: `stdin` or `input.txt`
Output file: `stdout` or `output.txt`
Time limit: 2 second(s)
Memory limit: 256 MiB

Last Christmas had been the best in recorded history for Lola. Santa had brought the little girl the most exciting present you could possibly imagine – a sequence of positive integers! The sequence has been standing in her playroom, in the original order, for almost a year now.

This year, he has topped even that. The new gift is an infinite supply of two arithmetic operators – plus (+) and times (×). Lola can now put one operator between each two consecutive numbers in the sequence and obtain an arithmetic expression!

Lola, though little, is already a bit sly and greedy, and she wants to maximize the value of the resulting expression. To restrain her somewhat (and to prevent the value from growing exponentially), Lola's parents are prohibiting her from putting two times (×) signs on both sides of any element of the sequence.

If Lola had paid attention in her kindergarten Algorithms class, she would know how to get the maximum possible value. However, while she knows maths and operator precedence and all that, algorithms were never really her strong suit... She is asking for your help.

Input

The first line of the input holds a single integer n ($1 \leq n \leq 1000$) – the length of the sequence. The second line holds n space-separated integers a_1, a_2, \dots, a_n – elements of the sequence ($1 \leq a_i \leq 1000$).

Output

Output one number – the maximum possible value of an expression Lola can get from the input sequence.

Examples

<code>stdin</code> or <code>input.txt</code>	<code>stdout</code> or <code>output.txt</code>
3 2 4 5	22
2 1 1	2
1 1000	1000
7 17 64 90 81 12 31 111	11831

Note

In the first sample testcase, the maximum value is $2 + 4 \times 5 = 22$. The expression $2 \times 4 \times 5$ is forbidden.

Problem D. In The Dark

Input file: `stdin` or `input.txt`
Output file: `stdout` or `output.txt`
Time limit: 2 second(s)
Memory limit: 256 MiB

It's Christmas night at EPFL, and the action begins. Santa's reindeer sleigh has just dumped him on the roof of Centre Midi, and from there he made his way inside through the chimney. Now he is totally and truly lost. After walking for some time in the dark, banging his belly against the walls, Santa gave up and is asking for your help.

You have managed to find a plan of the building in ASCII format and locate on it both Santa and the room where he is supposed to leave his presents. Now you want to annotate it with a shortest path that Santa should take to reach the room, and send it to his phone.

The plan is an $n \times n$ grid of cells. Santa can move between adjacent cells if they share a side. Some of the cells contain walls; although Santa has many talents, walking through walls is not one of them. He also cannot leave the grid.

Input

The first line of the input contains one integer n ($2 \leq n \leq 200$). The next n lines contain the plan of the building. Each line contains n characters. Each character is one of:

- A: the current position of Santa,
- B: the location of the room where Santa should leave the presents,
- W: a wall,
- N: a normal (non-wall) cell.

It is guaranteed that there is exactly one cell containing A, one cell containing B, and that there exists a path between them (which does not pass through a wall).

Output

Output the same $n \times n$ grid as given in the input, but change some of the characters N to P. The cells with characters A, B and P should together form a shortest path between A and B in the grid (which does not pass through a wall). If there are many such shortest paths, you may output any of them. Do not print anything else (e.g. extra whitespace on the beginning of the line). See the samples for clarity.

Examples

stdin or input.txt	stdout or output.txt
2 AB WW	AB WW
2 BN NA	BP NA
5 ANNNN WWWWN NNNNN NWWWW NNBNN	APPPP WWWWP PPPPP PWWWW PPBNN

Problem E. Grapes

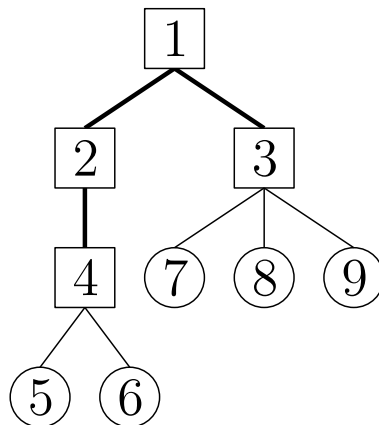
Input file: `stdin` or `input.txt`
Output file: `stdout` or `output.txt`
Time limit: 30 second(s)
Memory limit: 1024 MiB

Someone lately told Santa that grapes would make the perfect Christmas present for children. Indeed, kids love grapes, it's a cheap and healthy gift – what's not to like? So Santa went ahead and ordered a few megatons.

Now, little Miikka is one of the children who have received a bunch of grapes under the Christmas tree. Unfortunately, his parents are making him share it with his sister Lola, who also likes grapes very much.

Lola is rather predictable, so Miikka knows that as soon as he breaks one twig of the bunch, Lola will insist that, since she is younger, she should be allowed to choose one of the two obtained parts. And, of course, she will take the one with the larger number of grapes. Hence Miikka would like to choose the twig so as to guarantee that the other part (which he gets to keep) has as many grapes as possible. Help him determine how many he can guarantee for himself.

Miikka uses a rather peculiar terminology when talking about a bunch of grapes. Any such bunch has a tree structure: it consists of *nodes* and *twigs* (edges), and every twig directly connects two different nodes. Any two nodes are connected with a uniquely determined sequence of twigs. One node is distinguished and called the *root*. Nodes different from the root and such that there is exactly one twig connecting them to another node are the grapes (non-root leaves).



Input

The first line of input contains one integer n ($2 \leq n \leq 10^6$) denoting the number of nodes in the bunch. Nodes are numbered from 1 to n , where the root has number 1. The next $n - 1$ lines describe the twigs, each on a separate line. Each of these lines contains two integers a and b ($a \neq b$, $1 \leq a, b \leq n$) denoting that nodes with the corresponding numbers are directly connected by a twig.

Output

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

Examples

stdin or input.txt	stdout or output.txt
9 1 2 1 3 2 4 4 5 4 6 3 7 3 8 3 9	2
2 2 1	0

Note

The figure above shows the bunch from the first example testcase. Each grape is represented by a circle, and all other nodes are represented by squares. Segments connecting circles and squares represent twigs. Miikka can guarantee getting grapes with numbers 5 and 6 by breaking any of the twigs represented by bold segments. The remaining grapes with numbers 7, 8 and 9 go to Lola.

In the second testcase, recall that the root is not a grape. Therefore Miikka will not get much use out of his present.

Problem F. Beautiful Sleigh

Input file: `stdin` or `input.txt`
Output file: `stdout` or `output.txt`
Time limit: 5 second(s)
Memory limit: 256 MiB

You may have already noticed that Santa Claus is slightly eccentric. This year, he has decided that his sleigh has to be *beautiful*. And his concept of beauty is anything but standard.

The sleigh is described by a sequence of lowercase English letters which denote the colors of consecutive reindeer in the sleigh. Santa thinks that the sleigh is beautiful if this sequence can be divided into some number – greater than one – of consecutive subsequences which are all equal. For example, `ababab` and `aaa` are beautiful, whereas `abba` is not.

To make his sleigh beautiful, Santa may repaint some of the reindeer. Don't worry – no animals are harmed in this process. However, they will be displeased with the inconvenience (especially Rudolph). Santa has several painting methods at his disposal, each of which is described by a triple (α, β, γ) and allows Santa to paint one reindeer of color α with color β , while introducing γ units of displeasure to the sleigh. One reindeer may be painted over multiple times.

Help Santa plan his painting: compute the minimum displeasure of the sleigh.

Input

The first line of input holds two space-separated integers n and m – the number of reindeer in the sleigh and number of available painting methods, respectively ($2 \leq n \leq 10^5$, $1 \leq m \leq 10^5$). The second line contains n lowercase English letters which describe Santa's initial sleigh. The next m lines describe the painting methods. One painting method is described as $\alpha \beta \gamma$, where α and β are lowercase English letters and γ is a nonnegative integer ($0 \leq \gamma \leq 10^5$).

Output

Compute the minimum displeasure needed to make the sleigh beautiful, or output -1 if it is impossible to do so.

Examples

stdin or input.txt	stdout or output.txt
6 4 abcdba d a 3 a z 3 z c 2 a d 1	6

Note

In the sample testcase, the optimum solution is to repaint the first reindeer with color `d`, the sixth one with color `z`, and again the sixth one with color `c`.

Problem G. Elvish Game

Input file: `stdin` or `input.txt`
Output file: `stdout` or `output.txt`
Time limit: 2 second(s)
Memory limit: 256 MiB

For his visit in Lausanne this year, Santa has prepared 26 kinds of presents, denoted by lowercase English letters.

Santa also has an initial bag of presents described by a string of letters. For example, `aca` would describe a bag of two presents of the first kind and one present of the third kind. The order of the letters does not matter.

However, today his reindeer went on strike to protest against excessive repaintings. For this reason, he won't be able to bring more than one present of a kind. Now the contents of his initial bag will probably need to change.

Santa's operation is international, and just recently he has hired two new elves from Russia, called Dima and Grisha. To introduce them to the kind of peculiar challenges that they will be facing at their new job, he decided to have them play a game to determine the contents of the new bag.

The elves take turns, with Grisha starting. One turn is as follows: the current elf chooses two presents of the same kind from the current bag. These two presents are then replaced with a new one; the elf chooses the kind of the new present. The new kind can be any one of the 26 possible.

For example, if the game starts from the bag `aca`, then its contents after one move will be one of: `ca`, `cb`, `cc`, `cd`, ..., `cz`.

When all presents in the bag are of different kinds, no move is possible, and the current elf loses the game. Both elves only care about winning and they play optimally. Which one will win?

Input

The first and only line of input contains a single word, of length between 1 and 10^5 , describing Santa's initial bag.

Output

Output the name of the winning elf – Dima or Grisha.

Examples

<code>stdin</code> or <code>input.txt</code>	<code>stdout</code> or <code>output.txt</code>
<code>abc</code>	Dima
<code>aba</code>	Grisha

Problem H. Infinite Incantation

Input file: `stdin` or `input.txt`
Output file: `stdout` or `output.txt`
Time limit: 2 second(s)
Memory limit: 256 MiB

Santa has got himself locked in his bathroom (again). There is only one way for him to get out, and that is to invoke an Infinite Incantation of the bathroom-opening spell S .

In this digital age, this is done using a spellcasting grid mounted on the bathroom door. The grid is an $n \times m$ array of buttons labeled with lowercase English letters. The Incantation is performed by pressing a sequence of buttons; each two consecutive buttons in the sequence must be adjacent (which means sharing a side; a button is not adjacent to itself). Pressing a button generates the letter written on it.

One *stage* of the Incantation consists of a cyclic sequence of button presses, which means that the last button should be adjacent to the first. The stage is then repeated over and over, generating an infinite sequence of letters. The Incantation will be successful if this infinite sequence is equal to the infinite repetition of the spell S , which is a known sequence of lowercase English letters. (You might think that since the Incantation is Infinite, Santa will be stuck in the bathroom forever anyway. Don't worry: using his magical powers, he will press the first button in 1 second, the second one in $\frac{1}{2}$ seconds, the third one in $\frac{1}{4}$ seconds... He will be out in just two seconds.)

Help Santa find a description of a single cyclic stage which will get him out of the bathroom, or give him the bad news that it's impossible to do so.

Input

The first line of input holds three space-separated integers n , m and s – the numbers of rows and columns of the grid and the length of the spell, respectively ($2 \leq n, m \leq 200$, $1 \leq s \leq 200$). The next n lines hold m lowercase English letters each – the labels on the buttons. The last line holds s lowercase English letters – the spell S .

Output

If there is no way to free Santa, print **NO**.

Otherwise, on the first line, print **YES**. On the second line, print the length k of the stage which you have found. It should satisfy $2 \leq k \leq 10^7$. On the third line, print the number of row and the number of column of the button where the stage begins. On the fourth line, output the description of the stage, in the form of k letters **U**, **D**, **L** or **R**. These letters correspond to moving Santa's finger up, down, left or right respectively. After the last move, it should be back on the first button. The finger may not leave the spellcasting grid.

If there are multiple possible answers, you may output any of them.

Examples

stdin or input.txt	stdout or output.txt
3 2 4 aa bb cc abcb	YES 4 1 1 DDUU
3 3 5 abc dee ghi eeee	YES 2 2 2 RL
3 3 5 abc def ghi eeee	NO
3 6 16 abegme ckwass ichrdq egassemsdrawkcab	YES 16 1 3 RDRRULDDLULLLURR