Codeforces Round #426 (Div. 2)

A. The Useless Toy

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Walking through the streets of Marshmallow City, Slastyona have spotted some merchants selling a kind of useless toy which is very popular nowadays – caramel spinner! Wanting to join the craze, she has immediately bought the strange contraption.

Spinners in Sweetland have the form of V-shaped pieces of caramel. Each spinner can, well, spin around an invisible magic axis. At a specific point in time, a spinner can take 4 positions shown below (each one rotated 90 degrees relative to the previous, with the fourth one followed by the first one):

After the spinner was spun, it starts its rotation, which is described by a following algorithm: the spinner maintains its position for a second then majestically switches to the next position in clockwise or counter-clockwise order, depending on the direction the spinner was spun in.

Slastyona managed to have spinner rotating for exactly n seconds. Being fascinated by elegance of the process, she completely forgot the direction the spinner was spun in! Lucky for her, she managed to recall the starting position, and wants to deduct the direction given the information she knows. Help her do this.

Input

There are two characters in the first string – the starting and the ending position of a spinner. The position is encoded with one of the following characters: `v` (ASCII code 118, lowercase v), `<` (ASCII code 60), `^` (ASCII code 94) or `>` (ASCII code 62) (see the picture above for reference). Characters are separated by a single space.

In the second strings, a single number n is given ($0 \le n \le 10^9$) – the duration of the rotation.

It is guaranteed that the ending position of a spinner is a result of a n second spin in any of the directions, assuming the given starting position.

Output

Output `cw`, if the direction is clockwise, `ccw` – if counter-clockwise, and `undefined` otherwise.

Examples

input
^ >   1
output
cw

input
< ^   3
output
ccw

input
^ v   6

output
undefined

B. The Festive Evening

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

It's the end of July – the time when a festive evening is held at Jelly Castle! Guests from all over the kingdom gather here to discuss new trends in the world of confectionery. Yet some of the things discussed here are not supposed to be disclosed to the general public: the information can cause discord in the kingdom of Sweetland in case it turns out to reach the wrong hands. So it's a necessity to not let any uninvited guests in.

There are 26 entrances in Jelly Castle, enumerated with uppercase English letters from `A` to `Z`. Because of security measures, each guest is known to be assigned an entrance he should enter the castle through. The door of each entrance is opened right before the first guest's arrival and closed right after the arrival of the last guest that should enter the castle through this entrance. No two guests can enter the castle simultaneously.

For an entrance to be protected from possible intrusion, a candy guard should be assigned to it. There are k such guards in the castle, so if there are more than k opened doors, one of them is going to be left unguarded! Notice that a guard can't leave his post until the door he is assigned to is closed.

Slastyona had a suspicion that there could be uninvited guests at the evening. She knows the order in which the invited guests entered the castle, and wants you to help her check whether there was a moment when more than k doors were opened.

Input

Two integers are given in the first string: the number of guests n and the number of guards k ($1 \le n \le 10^6$, $1 \le k \le 26$).

In the second string, n uppercase English letters $s_1 s_2 \ldots s_n$ are given, where s_i is the entrance used by the i-th guest.

Output

Output «`YES`» if at least one door was unguarded during some time, and «`NO`» otherwise.

You can output each letter in arbitrary case (upper or lower).

Examples

input
5 1 AABBB
output
NO

input
5 1 ABABB
output
YES

Note

In the first sample case, the door A is opened right before the first guest's arrival and closed when the second guest enters the castle. The door B is opened right before the arrival of the third guest, and closed after the fifth one arrives. One guard can handle both doors, as the first one is closed before the second one is opened.

In the second sample case, the door B is opened before the second guest's arrival, but the only guard can't leave the door A unattended, as there is still one more guest that should enter the castle through this door.

C. The Meaningless Game

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Slastyona and her loyal dog Pushok are playing a meaningless *game* that is indeed very interesting.

The *game* consists of multiple *rounds*. Its rules are very simple: in each round, a natural number k is chosen. Then, the one who says (or barks) it faster than the other wins the *round*. After that, the winner's score is multiplied by k^2, and the loser's score is multiplied by k. In the beginning of the *game*, both Slastyona and Pushok have scores equal to one.

Unfortunately, Slastyona had lost her notepad where the history of all n *games* was recorded. She managed to recall the final results for each games, though, but all of her memories of them are vague. Help Slastyona verify their correctness, or, to put it another way, for each given pair of scores determine whether it was possible for a game to finish with such result or not.

Input

In the first string, the number of games n $(1 \le n \le 350000)$ is given.

Each *game* is represented by a pair of scores a, b $(1 \le a, b \le 10^9)$ – the results of Slastyona and Pushok, correspondingly.

Output

For each pair of scores, answer "`Yes`" if it's possible for a game to finish with given score, and "`No`" otherwise.

You can output each letter in arbitrary case (upper or lower).

Example

input
6 2 4 75 45 8 8 16 16 247 994 1000000000 1000000
output
Yes Yes Yes No No Yes

Note

First *game* might have been consisted of one round, in which the number 2 would have been chosen and Pushok would have won.

The second *game* needs exactly two rounds to finish with such result: in the first one, Slastyona would have said the number 5, and in the second one, Pushok would have barked the number 3.

D. The Bakery

time limit per test: 2.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Some time ago Slastyona the Sweetmaid decided to open her own bakery! She bought required ingredients and a wonder-oven which can bake several types of cakes, and opened the bakery.

Soon the expenses started to overcome the income, so Slastyona decided to study the sweets market. She learned it's profitable to pack cakes in boxes, and that the more **distinct** cake types a box contains (let's denote this number as the *value* of the box), the higher price it has.

She needs to change the production technology! The problem is that the oven chooses the cake types on its own and Slastyona can't affect it. However, she knows the types and order of n cakes the oven is going to bake today. Slastyona has to pack exactly k boxes with cakes today, and she has to put in each box several (at least one) cakes the oven produced one **right after another** (in other words, she has to put in a box a continuous segment of cakes).

Slastyona wants to maximize the total value of all boxes with cakes. Help her determine this maximum possible total value.

Input

The first line contains two integers n and k ($1 \le n \le 35000$, $1 \le k \le min(n, 50)$) – the number of cakes and the number of boxes, respectively.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$) – the types of cakes in the order the oven bakes them.

Output

Print the only integer – the maximum total value of all boxes with cakes.

Examples

input
4 1 1 2 2 1
output
2

input
7 2 1 3 3 1 4 4 4
output
5

input
8 3 7 7 8 7 7 8 1 7
output
6

Note

In the first example Slastyona has only one box. She has to put all cakes in it, so that there are two types of cakes in the box, so the value is equal to 2.

In the second example it is profitable to put the first two cakes in the first box, and all the rest in the second. There are two distinct types in the first box, and three in the second box then, so the total value is 5.

E. Ever-Hungry Krakozyabra

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Recently, a wild Krakozyabra appeared at Jelly Castle. It is, truth to be said, always eager to have something for dinner.

Its favorite meal is natural numbers (typically served with honey sauce), or, to be more precise, the zeros in their corresponding decimal representations. As for other digits, Krakozyabra dislikes them; moreover, they often cause it indigestion! So, as a necessary precaution, Krakozyabra prefers to sort the digits of a number in non-descending order before proceeding to feast. Then, the leading zeros of the resulting number are eaten and the remaining part is discarded as an *inedible tail*.

For example, if Krakozyabra is to have the number 57040 for dinner, its *inedible tail* would be the number 457.

Slastyona is not really fond of the idea of Krakozyabra living in her castle. Hovewer, her natural hospitality prevents her from leaving her guest without food. Slastyona has a range of natural numbers from L to R, which she is going to feed the guest with. Help her determine how many distinct *inedible tails* are going to be discarded by Krakozyabra by the end of the dinner.

Input

In the first and only string, the numbers L and R are given – the boundaries of the range $(1 \le L \le R \le 10^{18})$.

Output

Output the sole number – the answer for the problem.

Examples

input
1 10
output
9

input
40 57
output
17

input
157 165
output
9

Note

In the first sample case, the *inedible tails* are the numbers from 1 to 9. Note that 10 and 1 have the same *inedible tail* – the number 1.

In the second sample case, each number has a unique *inedible tail*, except for the pair $45, 54$. The answer to this sample case is going to be $(57 - 40 + 1) - 1 = 17$.