Codeforces Round #361 (Div. 2)

A. Mike and Cellphone

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

While swimming at the beach, Mike has accidentally dropped his cellphone into the water. There was no worry as he bought a cheap replacement phone with an old-fashioned keyboard. The keyboard has only ten digital equal-sized keys, located in the following way:

Together with his old phone, he lost all his contacts and now he can only remember the way his fingers moved when he put some number in. One can formally consider *finger movements* as a sequence of vectors connecting centers of keys pressed consecutively to put in a number. For example, the finger movements for number "586" are the same as finger movements for number "253":

Mike has already put in a number by his "finger memory" and started calling it, so he is now worrying, can he be sure that he is calling the correct number? In other words, is there any other number, that has the same finger movements?

Input

The first line of the input contains the only integer n $(1 \le n \le 9)$ — the number of digits in the phone number that Mike put in.

The second line contains the string consisting of n digits (characters from '0' to '9') representing the number that Mike put in.

Output

If there is no other phone number with the same finger movements and Mike can be sure he is calling the correct number, print "YES" (without quotes) in the only line.

Otherwise print "NO" (without quotes) in the first line.

Examples

input
```
3
586
```
output
```
NO
```

input
```
2
09
```
output
```
NO
```

input
```
9
123456789
```
output
```
YES
```

input
```
3
911
```
output
```
YES
```

Note

You can find the picture clarifying the first sample case in the statement above.

B. Mike and Shortcuts

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Recently, Mike was very busy with studying for exams and contests. Now he is going to chill a bit by doing some sight seeing in the city.

City consists of n intersections numbered from 1 to n. Mike starts walking from his house located at the intersection number 1 and goes along some sequence of intersections. Walking from intersection number i to intersection j requires $|i - j|$ units of energy. The *total energy* spent by Mike to visit a sequence of intersections $p_1 = 1, p_2, ..., p_k$ is equal to units of energy.

Of course, walking would be boring if there were no shortcuts. A *shortcut* is a special path that allows Mike walking from one intersection to another requiring only 1 unit of energy. There are exactly n shortcuts in Mike's city, the i^{th} of them allows walking from intersection i to intersection a_i ($i \le a_i \le a_{i+1}$) (but not in the opposite direction), thus there is exactly one shortcut starting at each intersection. Formally, if Mike chooses a sequence $p_1 = 1, p_2, ..., p_k$ then for each $1 \le i < k$ satisfying $p_{i+1} = a_{p_i}$ and $a_{p_i} \ne p_i$ Mike will spend **only 1 unit of energy** instead of $|p_i - p_{i+1}|$ walking from the intersection p_i to intersection p_{i+1}. For example, if Mike chooses a sequence $p_1 = 1, p_2 = a_{p_1}, p_3 = a_{p_2}, ..., p_k = a_{p_{k-1}}$, he spends exactly $k - 1$ units of total energy walking around them.

Before going on his adventure, Mike asks you to find the minimum amount of energy required to reach each of the intersections from his home. Formally, for each $1 \le i \le n$ Mike is interested in finding minimum possible total energy of some sequence $p_1 = 1, p_2, ..., p_k = i$.

Input

The first line contains an integer n $(1 \le n \le 200\,000)$ — the number of Mike's city intersection.

The second line contains n integers $a_1, a_2, ..., a_n$ $(i \le a_i \le n$, , describing shortcuts of Mike's city, allowing to walk from intersection i to intersection a_i using only 1 unit of energy. Please note that the shortcuts don't allow walking in opposite directions (from a_i to i).

Output

In the only line print n integers $m_1, m_2, ..., m_n$, where m_i denotes the least amount of total energy required to walk from intersection 1 to intersection i.

Examples

input

```
3
2 2 3
```

output

```
0 1 2
```

input

```
5
1 2 3 4 5
```

output

```
0 1 2 3 4
```

input

```
7
4 4 4 4 7 7 7
```

output

```
0 1 2 1 2 3 3
```

Note

In the first sample case desired sequences are:

$1: 1; m_1 = 0;$

$2: 1, 2; m_2 = 1;$

$3: 1, 3; m_3 = |3 - 1| = 2.$

In the second sample case the sequence for any intersection $1 < i$ is always $1, i$ and $m_i = |1 - i|$.

In the third sample case — consider the following intersection sequences:

$1: 1; m_1 = 0;$

$2: 1, 2; m_2 = |2 - 1| = 1;$

$3: 1, 4, 3; m_3 = 1 + |4 - 3| = 2;$

$4: 1, 4; m_4 = 1;$

5: 1, 4, 5; $m_5 = 1 + |4 - 5| = 2$;

6: 1, 4, 6; $m_6 = 1 + |4 - 6| = 3$;

7: 1, 4, 5, 7; $m_7 = 1 + |4 - 5| + 1 = 3$.

C. Mike and Chocolate Thieves

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bad news came to Mike's village, some thieves stole a bunch of chocolates from the local factory! Horrible!

Aside from loving sweet things, thieves from this area are known to be very greedy. So after a thief takes his number of chocolates for himself, the next thief will take exactly k times more than the previous one. The value of k ($k > 1$) is a secret integer known only to them. It is also known that each thief's bag can carry at most n chocolates (if they intend to take more, the deal is cancelled) and that there were **exactly four** thieves involved.

Sadly, only the thieves know the value of n, but rumours say that the numbers of ways they could have taken the chocolates (for a fixed n, but not fixed k) is m. Two ways are considered different if one of the thieves (they should be numbered in the order they take chocolates) took different number of chocolates in them.

Mike want to track the thieves down, so he wants to know what their bags are and value of n will help him in that. Please find **the smallest possible** value of n or tell him that the rumors are false and there is no such n.

Input

The single line of input contains the integer m $(1 \le m \le 10^{15})$ — the number of ways the thieves might steal the chocolates, as rumours say.

Output

Print the only integer n — the maximum amount of chocolates that thieves' bags can carry. If there are more than one n satisfying the rumors, **print the smallest one**.

If there is no such n for a false-rumoured m, print -1.

Examples

input
```
1
```
output
```
8
```

input
```
8
```
output
```
54
```

input
```
10
```
output
```
-1
```

Note

In the first sample case the smallest n that leads to exactly one way of stealing chocolates is $n = 8$, whereas the amounts of stealed chocolates are $(1, 2, 4, 8)$ (the number of chocolates stolen by each of the thieves).

In the second sample case the smallest n that leads to exactly 8 ways is $n = 54$ with the possibilities: $(1, 2, 4, 8)$, $(1, 3, 9, 27)$, $(2, 4, 8, 16)$, $(2, 6, 18, 54)$, $(3, 6, 12, 24)$, $(4, 8, 16, 32)$, $(5, 10, 20, 40)$, $(6, 12, 24, 48)$.

There is no n leading to exactly 10 ways of stealing chocolates in the third sample case.

D. Friends and Subsequences

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Mike and !Mike are old childhood rivals, they are opposite in everything they do, except programming. Today they have a problem they cannot solve on their own, but together (with you) — who knows?

Every one of them has an integer sequences a and b of length n. Being given a query of the form of pair of integers (l, r), Mike can instantly tell the value of while !Mike can instantly tell the value of .

Now suppose a robot (you!) asks them all possible different queries of pairs of integers (l, r) $(1 \le l \le r \le n)$ (so he will make exactly $n(n + 1) / 2$ queries) and counts how many times their answers coincide, thus for how many pairs is satisfied.

How many occasions will the robot count?

Input

The first line contains only integer n ($1 \le n \le 200\,000$).

The second line contains n integer numbers $a_1, a_2, ..., a_n$ ($-10^9 \le a_i \le 10^9$) — the sequence a.

The third line contains n integer numbers $b_1, b_2, ..., b_n$ ($-10^9 \le b_i \le 10^9$) — the sequence b.

Output

Print the only integer number — the number of occasions the robot will count, thus for how many pairs is satisfied.

Examples

input

```
6
1 2 3 2 1 4
6 7 1 2 3 2
```

output

```
2
```

input

```
3
3 3 3
1 1 1
```

output

```
0
```

Note

The occasions in the first sample case are:

1. $l = 4, r = 4$ since $max\{2\} = min\{2\}$.

2. $l = 4, r = 5$ since $max\{2, 1\} = min\{2, 3\}$.

There are no occasions in the second sample case since Mike will answer 3 to any query pair, but !Mike will always answer 1.

E. Mike and Geometry Problem

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Mike wants to prepare for IMO but he doesn't know geometry, so his teacher gave him an interesting geometry problem. Let's define $f([l, r]) = r - l + 1$ to be the number of integer points in the segment $[l, r]$ with $l \leq r$ (say that). You are given two integers n and k and n closed intervals $[l_i, r_i]$ on OX axis and you have to find:

In other words, you should find the sum of the number of integer points in the intersection of any k of the segments.

As the answer may be very large, output it modulo 1000000007 $(10^9 + 7)$.

Mike can't solve this problem so he needs your help. You will help him, won't you?

Input

The first line contains two integers n and k $(1 \leq k \leq n \leq 200\ 000)$ — the number of segments and the number of segments in intersection groups respectively.

Then n lines follow, the i-th line contains two integers l_i, r_i $(-10^9 \leq l_i \leq r_i \leq 10^9)$, describing i-th segment bounds.

Output

Print one integer number — the answer to Mike's problem modulo 1000000007 $(10^9 + 7)$ in the only line.

Examples

input

```
3 2
1 2
1 3
2 3
```

output

```
5
```

input

```
3 3
1 3
1 3
1 3
```

output

```
3
```

input

```
3 1
1 2
2 3
3 4
```

output

```
6
```

Note

In the first example:

;

;

.

So the answer is $2 + 1 + 2 = 5$.