

## Codeforces Round #189 (Div. 1)

### A. Malek Dance Club

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

As a tradition, every year before IOI all the members of Natalia Fan Club are invited to Malek Dance Club to have a fun night together. Malek Dance Club has  $2^n$  members and coincidentally Natalia Fan Club also has  $2^n$  members. Each member of MDC is assigned a unique id  $i$  from 0 to  $2^n - 1$ . The same holds for each member of NFC.

One of the parts of this tradition is one by one dance, where each member of MDC dances with a member of NFC. A dance pair is a pair of numbers  $(a, b)$  such that member  $a$  from MDC dances with member  $b$  from NFC.

The complexity of a pairs' assignment is the number of pairs of dancing pairs  $(a, b)$  and  $(c, d)$  such that  $a < c$  and  $b > d$ .

You are given a binary number of length  $n$  named  $x$ . We know that member  $i$  from MDC dances with member  $i \oplus x$  from NFC. Your task is to calculate the complexity of this assignment modulo  $1000000007$  ( $10^9 + 7$ ).

Expression  $x \oplus y$  denotes applying «XOR» to numbers  $x$  and  $y$ . This operation exists in all modern programming languages, for example, in C++ and Java it denotes as «^», in Pascal — «xor».

#### Input

The first line of input contains a binary number  $x$  of length  $n$ , ( $1 \leq n \leq 100$ ).

This number may contain leading zeros.

#### Output

Print the complexity of the given dance assignment modulo  $1000000007$  ( $10^9 + 7$ ).

#### Sample test(s)

input	
11	
output	
6	
input	
01	
output	
2	
input	
1	
output	
1	

## B. Psychos in a Line

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

There are  $n$  psychos standing in a line. Each psycho is assigned a unique integer from  $1$  to  $n$ . At each step every psycho who has an id greater than the psycho to his right (if exists) kills his right neighbor in the line. Note that a psycho might kill and get killed at the same step.

You're given the initial arrangement of the psychos in the line. Calculate how many steps are needed to the moment of time such, that nobody kills his neighbor after that moment. Look notes to understand the statement more precise.

### Input

The first line of input contains integer  $n$  denoting the number of psychos, ( $1 \leq n \leq 10^5$ ). In the second line there will be a list of  $n$  space separated distinct integers each in range  $1$  to  $n$ , inclusive — ids of the psychos in the line from left to right.

### Output

Print the number of steps, so that the line remains the same afterward.

### Sample test(s)

input
10 10 9 7 8 6 5 3 4 2 1
output
2
input
6 1 2 3 4 5 6
output
0

### Note

In the first sample line of the psychos transforms as follows:  $[10\ 9\ 7\ 8\ 6\ 5\ 3\ 4\ 2\ 1] \rightarrow [10\ 8\ 4] \rightarrow [10]$ . So, there are two steps.

## C. Kalila and Dimna in the Logging Industry

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

Kalila and Dimna are two jackals living in a huge jungle. One day they decided to join a logging factory in order to make money.

The manager of logging factory wants them to go to the jungle and cut  $n$  trees with heights  $a_1, a_2, \dots, a_n$ . They bought a chain saw from a shop. Each time they use the chain saw on the tree number  $i$ , they can decrease the height of this tree by one unit. Each time that Kalila and Dimna use the chain saw, they need to recharge it. Cost of charging depends on the id of the trees which have been cut completely (a tree is cut completely if its height equal to 0). If the maximum id of a tree which has been cut completely is  $i$  (the tree that have height  $a_i$  in the beginning), then the cost of charging the chain saw would be  $b_i$ . If no tree is cut completely, Kalila and Dimna cannot charge the chain saw. The chainsaw is charged in the beginning. We know that for each  $i < j$ ,  $a_i < a_j$  and  $b_i > b_j$  and also  $b_n = 0$  and  $a_1 = 1$ . Kalila and Dimna want to cut all the trees completely, with minimum cost.

They want you to help them! Will you?

### Input

The first line of input contains an integer  $n$  ( $1 \leq n \leq 10^5$ ). The second line of input contains  $n$  integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ). The third line of input contains  $n$  integers  $b_1, b_2, \dots, b_n$  ( $0 \leq b_i \leq 10^9$ ).

It's guaranteed that  $a_1 = 1$ ,  $b_n = 0$ ,  $a_1 < a_2 < \dots < a_n$  and  $b_1 > b_2 > \dots > b_n$ .

### Output

The only line of output must contain the minimum cost of cutting all the trees completely.

Please, do not write the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

### Sample test(s)

input
5 1 2 3 4 5 5 4 3 2 0
output
25
input
6 1 2 3 10 20 30 6 5 4 3 2 0
output
138

## D. Have You Ever Heard About the Word?

time limit per test: 6 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

A *substring* of a string is a contiguous subsequence of that string. So, string `bca` is substring of string `abcabc`, but string `cc` is not.

A *repeating block* is a string formed by concatenating some string with itself. So, string `abcabc` is a repeating block, but strings `abcabd`, `ababab` are not.

You've got a sequence of Latin characters (string). At each step you find the shortest substring that is a repeating block, if there exists more than one you must choose the leftmost. As the substring is of form  $XX$  ( $X$  — some string) you replace this substring with  $X$ , in other words you delete one of the  $X$  substrings in the substring. You repeat this process until there remains no repeating block in the string.

How would the final string looks like? Look at the sample explanation to understand the statement more precise.

### Input

In the first line of input you're given a string of small Latin characters with length between 1 to 50000, inclusive.

### Output

Print the final string after applying changes.

### Sample test(s)

input
<code>abccabc</code>
output
<code>abc</code>
input
<code>aaaabaaab</code>
output
<code>ab</code>
input
<code>birdbirdbirdistheword</code>
output
<code>birdistheword</code>

### Note

At the first sample the string transforms as follows: `abccabc`  $\rightarrow$  `abcabc`  $\rightarrow$  `abc`.

At the second sample the string transforms as follows: `aaaabaaab`  $\rightarrow$  `aaabaaab`  $\rightarrow$  `aabaaab`  $\rightarrow$  `abaaab`  $\rightarrow$  `abaab`  $\rightarrow$  `abab`  $\rightarrow$  `ab`.

## E. Ping-Pong

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

In this problem at each moment you have a set of intervals. You can move from interval  $(a, b)$  from our set to interval  $(c, d)$  from our set if and only if  $c < a < d$  or  $c < b < d$ . Also there is a path from interval  $I_1$  from our set to interval  $I_2$  from our set if there is a sequence of successive moves starting from  $I_1$  so that we can reach  $I_2$ .

Your program should handle the queries of the following two types:

1. "1 x y" ( $x < y$ ) — add the new interval  $(x, y)$  to the set of intervals. The length of the new interval is guaranteed to be strictly greater than all the previous intervals.
2. "2 a b" ( $a \neq b$ ) — answer the question: is there a path from  $a$ -th (one-based) added interval to  $b$ -th (one-based) added interval?

Answer all the queries. Note, that initially you have an empty set of intervals.

### Input

The first line of the input contains integer  $n$  denoting the number of queries, ( $1 \leq n \leq 10^5$ ). Each of the following lines contains a query as described above. All numbers in the input are integers and don't exceed  $10^9$  by their absolute value.

It's guaranteed that all queries are correct.

### Output

For each query of the second type print "YES" or "NO" on a separate line depending on the answer.

### Sample test(s)

input
5 1 1 5 1 5 11 2 1 2 1 2 9 2 1 2
output
NO YES