

Codeforces Round #136 (Div. 2)

A. Little Elephant and Function

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

The Little Elephant enjoys recursive functions.

This time he enjoys the sorting function. Let a is a permutation of an integers from 1 to n , inclusive, and a_i denotes the i -th element of the permutation. The Little Elephant's recursive function $f(x)$, that sorts the first x permutation's elements, works as follows:

- If $x = 1$, exit the function.
- Otherwise, call $f(x - 1)$, and then make $swap(a_{x-1}, a_x)$ (swap the x -th and $(x - 1)$ -th elements of a).

The Little Elephant's teacher believes that this function does not work correctly. But that-be do not get an F, the Little Elephant wants to show the performance of its function. Help him, find a permutation of numbers from 1 to n , such that after performing the Little Elephant's function (that is call $f(n)$), the permutation will be sorted in ascending order.

Input

A single line contains integer n ($1 \leq n \leq 1000$) — the size of permutation.

Output

In a single line print n distinct integers from 1 to n — the required permutation. Numbers in a line should be separated by spaces.

It is guaranteed that the answer exists.

Sample test(s)

input
1
output
1
input
2
output
2 1

B. Little Elephant and Numbers

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

The Little Elephant loves numbers.

He has a positive integer x . The Little Elephant wants to find the number of positive integers d , such that d is the divisor of x , and x and d have at least one common (the same) digit in their decimal representations.

Help the Little Elephant to find the described number.

Input

A single line contains a single integer x ($1 \leq x \leq 10^9$).

Output

In a single line print an integer — the answer to the problem.

Sample test(s)

input
1
output
1
input
10
output
2

C. Little Elephant and Problem

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

The Little Elephant has got a problem — somebody has been touching his sorted by non-decreasing array a of length n and possibly swapped some elements of the array.

The Little Elephant doesn't want to call the police until he understands if he could have accidentally changed the array himself. He thinks that he could have accidentally changed array a , only if array a can be sorted in no more than one operation of swapping elements (not necessarily adjacent). That is, the Little Elephant could have accidentally swapped some two elements.

Help the Little Elephant, determine if he could have accidentally changed the array a , sorted by non-decreasing, himself.

Input

The first line contains a single integer n ($2 \leq n \leq 10^5$) — the size of array a . The next line contains n positive integers, separated by single spaces and not exceeding 10^9 , — array a .

Note that the elements of the array are not necessarily distinct numbers.

Output

In a single line print "YES" (without the quotes) if the Little Elephant could have accidentally changed the array himself, and "NO" (without the quotes) otherwise.

Sample test(s)

input
2 1 2
output
YES
input
3 3 2 1
output
YES
input
4 4 3 2 1
output
NO

Note

In the first sample the array has already been sorted, so to sort it, we need 0 swap operations, that is not more than 1. Thus, the answer is "YES".

In the second sample we can sort the array if we swap elements 1 and 3, so we need 1 swap operation to sort the array. Thus, the answer is "YES".

In the third sample we can't sort the array in more than one swap operation, so the answer is "NO".

D. Little Elephant and Array

time limit per test: 4 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

The Little Elephant loves playing with arrays. He has array a , consisting of n positive integers, indexed from 1 to n . Let's denote the number with index i as a_i .

Additionally the Little Elephant has m queries to the array, each query is characterised by a pair of integers l_j and r_j ($1 \leq l_j \leq r_j \leq n$). For each query l_j, r_j the Little Elephant has to count, how many numbers x exist, such that number x occurs exactly x times among numbers $a_{l_j}, a_{l_j+1}, \dots, a_{r_j}$.

Help the Little Elephant to count the answers to all queries.

Input

The first line contains two space-separated integers n and m ($1 \leq n, m \leq 10^5$) — the size of array a and the number of queries to it. The next line contains n space-separated positive integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$). Next m lines contain descriptions of queries, one per line. The j -th of these lines contains the description of the j -th query as two space-separated integers l_j and r_j ($1 \leq l_j \leq r_j \leq n$).

Output

In m lines print m integers — the answers to the queries. The j -th line should contain the answer to the j -th query.

Sample test(s)

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

E. Little Elephant and Shifts

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

The Little Elephant has two permutations a and b of length n , consisting of numbers from 1 to n , inclusive. Let's denote the i -th ($1 \leq i \leq n$) element of the permutation a as a_i , the j -th ($1 \leq j \leq n$) element of the permutation b — as b_j .

The *distance* between permutations a and b is the minimum absolute value of the difference between the positions of the occurrences of some number in a and in b . More formally, it's such minimum $|i - j|$, that $a_i = b_j$.

A *cyclic shift* number i ($1 \leq i \leq n$) of permutation b consisting from n elements is a permutation $b_i b_{i+1} \dots b_n b_1 b_2 \dots b_{i-1}$. Overall a permutation has n cyclic shifts.

The Little Elephant wonders, for all cyclic shifts of permutation b , what is the distance between the cyclic shift and permutation a ?

Input

The first line contains a single integer n ($1 \leq n \leq 10^5$) — the size of the permutations. The second line contains permutation a as n distinct numbers from 1 to n , inclusive. The numbers are separated with single spaces. The third line contains permutation b in the same format.

Output

In n lines print n integers — the answers for cyclic shifts. Print the answers to the shifts in the order of the shifts' numeration in permutation b , that is, first for the 1-st cyclic shift, then for the 2-nd, and so on.

Sample test(s)

input
2 1 2 2 1
output
1 0

input
4 2 1 3 4 3 4 2 1
output
2 1 0 1