Codeforces Round #424 (Div. 2, rated, based on VK Cup Finals)

A. Unimodal Array

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

Array of integers is *unimodal*, if:

- it is strictly increasing in the beginning;
- after that it is constant;
- after that it is strictly decreasing.

The first block (increasing) and the last block (decreasing) may be absent. It is allowed that both of this blocks are absent.

For example, the following three arrays are unimodal: $[5, 7, 11, 11, 2, 1]$, $[4, 4, 2]$, $[7]$, but the following three are not unimodal: $[5, 5, 6, 6, 1]$, $[1, 2, 1, 2]$, $[4, 5, 5, 6]$.

Write a program that checks if an array is unimodal.

Input

The first line contains integer n ($1 \le n \le 100$) — the number of elements in the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 1\,000$) — the elements of the array.

Output

Print "YES" if the given array is unimodal. Otherwise, print "NO".

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

Examples

input
6 1 5 5 5 4 2
output
YES

input
5 10 20 30 20 10
output
YES

input
4 1 2 1 2
output
NO

input
7 3 3 3 3 3 3 3
output
YES

Note

In the first example the array is unimodal, because it is strictly increasing in the beginning (from position 1 to position 2, inclusively), that it is constant (from position 2 to position 4, inclusively) and then it is strictly decreasing (from position 4 to position 6, inclusively).

B. Keyboard Layouts

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

There are two popular keyboard layouts in Berland, they differ only in letters positions. All the other keys are the same. In Berland they use alphabet with 26 letters which coincides with English alphabet.

You are given two strings consisting of 26 distinct letters each: all keys of the first and the second layouts in the same order.

You are also given some text consisting of small and capital English letters and digits. It is known that it was typed in the first layout, but the writer intended to type it in the second layout. Print the text if the same keys were pressed in the second layout.

Since all keys but letters are the same in both layouts, the capitalization of the letters should remain the same, as well as all other characters.

Input

The first line contains a string of length 26 consisting of distinct lowercase English letters. This is the first layout.

The second line contains a string of length 26 consisting of distinct lowercase English letters. This is the second layout.

The third line contains a non-empty string s consisting of lowercase and uppercase English letters and digits. This is the text typed in the first layout. The length of s does not exceed 1000.

Output

Print the text if the same keys were pressed in the second layout.

Examples

input
qwertyuiopasdfghjklzxcvbnm veamhjsgqocnrbfxdtwkylupzi TwccpQZAvb2017
output
HelloVKCup2017

input
mnbvcxzlkjhgfdsapoiuytrewq asdfghjklqwertyuiopzxcvbnm 7abaCABAABAcaba7
output
7uduGUDUUDUgudu7

C. Jury Marks

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

Polycarp watched TV-show where k jury members one by one rated a participant by adding him a certain number of points (may be negative, i. e. points were subtracted). Initially the participant had some score, and each the marks were one by one added to his score. It is known that the i-th jury member gave a_i points.

Polycarp does not remember how many points the participant had before this k marks were given, but he remembers that among the scores announced after each of the k judges rated the participant there were n ($n \le k$) values $b_1, b_2, \ldots, b_n$ (it is guaranteed that all values b_j are distinct). It is possible that Polycarp remembers not all of the scores announced, i. e. $n < k$. Note that the initial score wasn't announced.

Your task is to determine the number of options for the score the participant could have before the judges rated the participant.

Input

The first line contains two integers k and n ($1 \le n \le k \le 2\,000$) — the number of jury members and the number of scores Polycarp remembers.

The second line contains k integers $a_1, a_2, \ldots, a_k$ ($-2\,000 \le a_i \le 2\,000$) — jury's marks in chronological order.

The third line contains n **distinct** integers $b_1, b_2, \ldots, b_n$ ($-4\,000\,000 \le b_j \le 4\,000\,000$) — the values of points Polycarp remembers. Note that these values are not necessarily given in chronological order.

Output

Print the number of options for the score the participant could have before the judges rated the participant. If Polycarp messes something up and there is no options, print "0" (without quotes).

Examples

input
4 1 -5 5 0 20 10
output
3

input
2 2 -2000 -2000 3998000 4000000
output
1

Note

The answer for the first example is 3 because initially the participant could have -10, 10 or 15 points.

In the second example there is only one correct initial score equaling to $4\,002\,000$.

D. Office Keys

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

There are n people and k keys on a straight line. Every person wants to get to the office which is located on the line as well. To do that, he needs to reach some point with a key, take the key and then go to the office. Once a key is taken by somebody, it couldn't be taken by anybody else.

You are to determine the minimum time needed for all n people to get to the office with keys. Assume that people move a unit distance per 1 second. If two people reach a key at the same time, only one of them can take the key. A person can pass through a point with a key without taking it.

Input

The first line contains three integers n, k and p ($1 \le n \le 1\,000$, $n \le k \le 2\,000$, $1 \le p \le 10^9$) — the number of people, the number of keys and the office location.

The second line contains n distinct integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — positions in which people are located initially. The positions are given in arbitrary order.

The third line contains k distinct integers $b_1, b_2, \ldots, b_k$ ($1 \le b_j \le 10^9$) — positions of the keys. The positions are given in arbitrary order.

Note that there can't be more than one person or more than one key in the same point. A person and a key can be located in the same point.

Output

Print the minimum time (in seconds) needed for all n to reach the office with keys.

Examples

input
2 4 50 20 100 60 10 40 80
output
50

input
1 2 10 11 15 7
output
7

Note

In the first example the person located at point 20 should take the key located at point 40 and go with it to the office located at point 50. He spends 30 seconds. The person located at point 100 can take the key located at point 80 and go to the office with it. He spends 50 seconds. Thus, after 50 seconds everybody is in office with keys.

E. Cards Sorting

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

Vasily has a deck of cards consisting of n cards. There is an integer on each of the cards, this integer is between 1 and $100\,000$, inclusive. It is possible that some cards have the same integers on them.

Vasily decided to sort the cards. To do this, he repeatedly takes the top card from the deck, and if the number on it equals the minimum number written on the cards in the deck, then he places the card away. Otherwise, he puts it under the deck and takes the next card from the top, and so on. The process ends as soon as there are no cards in the deck. You can assume that Vasily always knows the minimum number written on some card in the remaining deck, but doesn't know where this card (or these cards) is.

You are to determine the total number of times Vasily takes the top card from the deck.

Input

The first line contains single integer n ($1 \le n \le 100\,000$) — the number of cards in the deck.

The second line contains a sequence of n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 100\,000$), where a_i is the number written on the i-th from top card in the deck.

Output

Print the total number of times Vasily takes the top card from the deck.

Examples

input
4 6 3 1 2
output
7

input
1 1000
output
1

input
7 3 3 3 3 3 3 3
output
7

Note

In the first example Vasily at first looks at the card with number 6 on it, puts it under the deck, then on the card with number 3, puts it under the deck, and then on the card with number 1. He places away the card with 1, because the number written on it is the minimum among the remaining cards. After that the cards from top to bottom are $[2, 6, 3]$. Then Vasily looks at the top card with number 2 and puts it away. After that the cards from top to bottom are $[6, 3]$. Then Vasily looks at card 6, puts it under the deck, then at card 3 and puts it away. Then there is only one card with number 6 on it, and Vasily looks at it and puts it away. Thus, in total Vasily looks at 7 cards.

F. Bamboo Partition

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

Vladimir wants to modernize partitions in his office. To make the office more comfortable he decided to remove a partition and plant several bamboos in a row. He thinks it would be nice if there are n bamboos in a row, and the i-th from the left is a_i meters high.

Vladimir has just planted n bamboos in a row, each of which has height 0 meters right now, but they grow 1 meter each day. In order to make the partition nice Vladimir can cut each bamboo once at any height (no greater that the height of the bamboo), and then the bamboo will stop growing.

Vladimir wants to check the bamboos each d days (i.e. d days after he planted, then after $2d$ days and so on), and cut the bamboos that reached the required height. Vladimir wants the total length of bamboo parts he will cut off to be no greater than k meters.

What is the maximum value d he can choose so that he can achieve what he wants without cutting off more than k meters of bamboo?

Input

The first line contains two integers n and k ($1 \le n \le 100$, $1 \le k \le 10^{11}$) — the number of bamboos and the maximum total length of cut parts, in meters.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the required heights of bamboos, in meters.

Output

Print a single integer — the maximum value of d such that Vladimir can reach his goal.

Examples

input
3 4 1 3 5
output
3

input
3 40 10 30 50
output
32

Note

In the first example Vladimir can check bamboos each 3 days. Then he will cut the first and the second bamboos after 3 days, and the third bamboo after 6 days. The total length of cut parts is $2 + 0 + 1 = 3$ meters.

Codeforces (c) Copyright 2010-2017 Mike Mirzayanov
The only programming contests Web 2.0 platform