Codeforces Round #201 (Div. 2)

A. Difference Row

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

You want to arrange n integers $a_1, a_2, ..., a_n$ in some order in a row. Let's define the value of an arrangement as the sum of differences between all pairs of adjacent integers.

More formally, let's denote some arrangement as a sequence of integers $x_1, x_2, ..., x_n$, where sequence x is a permutation of sequence a. The value of such an arrangement is $(x_1 - x_2) + (x_2 - x_3) + ... + (x_{n-1} - x_n)$.

Find the largest possible value of an arrangement. Then, output the lexicographically smallest sequence x that corresponds to an arrangement of the largest possible value.

Input

The first line of the input contains integer n ($2 \le n \le 100$). The second line contains n space-separated integers $a_1, a_2, ..., a_n$ ($|a_i| \le 1000$).

Output

Print the required sequence $x_1, x_2, ..., x_n$. Sequence x should be the lexicographically smallest permutation of a that corresponds to an arrangement of the largest possible value.

Examples

input
5 100 -100 50 0 -50
output
100 -50 0 50 -100

Note

In the sample test case, the value of the output arrangement is $(100 - (-50)) + ((-50) - 0) + (0 - 50) + (50 - (-100)) = 200$. No other arrangement has a larger value, and among all arrangements with the value of 200, the output arrangement is the lexicographically smallest one.

Sequence $x_1, x_2, ..., x_p$ is *lexicographically smaller* than sequence $y_1, y_2, ..., y_p$ if there exists an integer r $(0 \le r < p)$ such that $x_1 = y_1, x_2 = y_2, ..., x_r = y_r$ and $x_{r+1} < y_{r+1}$.

B. Fixed Points

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

A permutation of length n is an integer sequence such that each integer from 0 to $(n - 1)$ appears exactly once in it. For example, sequence $[0, 2, 1]$ is a permutation of length 3 while both $[0, 2, 2]$ and $[1, 2, 3]$ are not.

A fixed point of a function is a point that is mapped to itself by the function. A permutation can be regarded as a bijective function. We'll get a definition of a fixed point in a permutation. An integer i is a fixed point of permutation $a_0, a_1, \ldots, a_{n-1}$ if and only if $a_i = i$. For example, permutation $[0, 2, 1]$ has 1 fixed point and permutation $[0, 1, 2]$ has 3 fixed points.

You are given permutation a. You are allowed to swap two elements of the permutation at most once. Your task is to maximize the number of fixed points in the resulting permutation. Note that you are allowed to make at most one swap operation.

Input

The first line contains a single integer n ($1 \le n \le 10^5$). The second line contains n integers $a_0, a_1, \ldots, a_{n-1}$ — the given permutation.

Output

Print a single integer — the maximum possible number of fixed points in the permutation after at most one swap operation.

Examples

input
5 0 1 3 4 2
output
3

C. Alice and Bob

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

It is so boring in the summer holiday, isn't it? So Alice and Bob have invented a new game to play. The rules are as follows. First, they get a set of n distinct integers. And then they take turns to make the following moves. During each move, either Alice or Bob (the player whose turn is the current) can choose two distinct integers x and y from the set, such that the set doesn't contain their absolute difference $|x - y|$. Then this player adds integer $|x - y|$ to the set (so, the size of the set increases by one).

If the current player has no valid move, he (or she) loses the game. The question is who will finally win the game if both players play optimally. Remember that Alice always moves first.

Input

The first line contains an integer n ($2 \le n \le 100$) — the initial number of elements in the set. The second line contains n distinct space-separated integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the elements of the set.

Output

Print a single line with the winner's name. If Alice wins print "Alice", otherwise print "Bob" (without quotes).

Examples

input
2 2 3
output
Alice

input
2 5 3
output
Alice

input
3 5 6 7
output
Bob

Note

Consider the first test sample. Alice moves first, and the only move she can do is to choose 2 and 3, then to add 1 to the set. Next Bob moves, there is no valid move anymore, so the winner is Alice.

D. Lucky Common Subsequence

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

In mathematics, a *subsequence* is a sequence that can be derived from another sequence by deleting some elements without changing the order of the remaining elements. For example, the sequence BDF is a subsequence of ABCDEF. A *substring* of a string is a continuous subsequence of the string. For example, BCD is a substring of ABCDEF.

You are given two strings s_1, s_2 and another string called $virus$. Your task is to find the longest common subsequence of s_1 and s_2, such that it doesn't contain $virus$ as a substring.

Input

The input contains three strings in three separate lines: s_1, s_2 and $virus$ ($1 \leq |s_1|, |s_2|, |virus| \leq 100$). Each string consists only of uppercase English letters.

Output

Output the longest common subsequence of s_1 and s_2 without $virus$ as a substring. If there are multiple answers, any of them will be accepted.

If there is no valid common subsequence, output 0.

Examples

input
AJKEQSLOBSROFGZ OVGURWZLWVLUXTH OZ
output
ORZ

input
AA A A
output
0

E. Number Transformation II

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

You are given a sequence of positive integers $x_1, x_2, ..., x_n$ and two non-negative integers a and b. Your task is to transform a into b. To do that, you can perform the following moves:

- subtract 1 from the current a;
- subtract a mod x_i $(1 \le i \le n)$ from the current a.

Operation a mod x_i means taking the remainder after division of number a by number x_i.

Now you want to know the minimum number of moves needed to transform a into b.

Input

The first line contains a single integer n ($1 \le n \le 10^5$). The second line contains n space-separated integers $x_1, x_2, ..., x_n$ ($2 \le x_i \le 10^9$). The third line contains two integers a and b ($0 \le b \le a \le 10^9$, $a - b \le 10^6$).

Output

Print a single integer — the required minimum number of moves needed to transform number a into number b.

Examples

input
3 3 4 5 30 17
output
6

input
3 5 6 7 1000 200
output
206