C: Chinese remainder theorem

Memory Limit : **256 MB**

Johnny is a computer science student. This semester he became well versed with the Chinese Remainder Theorem. While waiting for a next lecture he heard that Maggie complained that she cannot solve her homework; as he heard the familiar words "modulo" and "system of equations" he immediately offered his help to the damsel in distress. It turns out that Maggie's task is much different than those that Johnny is accustomed to solve, it is of the following form:

$$\begin{cases} a_1 & \equiv & b_1 & (\mathrm{mod}\ m) \\ a_2 & \equiv & b_2 & (\mathrm{mod}\ m) \\ \vdots & \vdots & & \vdots \\ a_n & \equiv & b_n & (\mathrm{mod}\ m) \end{cases}$$

(where $\equiv$ means equivalence modulo) and for the given $a_1, b_1, \ldots, a_n, b_n$ Maggie should compute the largest m such that all of the equations hold. Maggie already started processing the equations and she ensured that $a_i \geqslant b_i$ for each i. Johhny cannot fail and lose his face – help him to solve the task.

Input

The first line of the input contains a single natural number n ($1 \leqslant n \leqslant 100\,000$), denoting the number of equations.

The second line contains n integers $a_1, a_2, \ldots, a_n$, each separated by a single space, these are the numbers on the left-hand sides of consecutive equations.

The third and last line contains n integers $b_1, b_2, \ldots, b_n$, each separated by a single space, these are the numbers on the right-hand sides of consecutive equations.

The inequality $0 \leqslant b_i \leqslant a_i \leqslant 10^{18}$ holds for each i ($1 \leqslant i \leqslant n$). The system of equations is nontrivial: $a_i \neq b_i$ holds for some i ($1 \leqslant i \leqslant n$).

Output

You should write a single natural number in the first and only line of the output – the largest m for which the given system of equations is satisfied.

Example

Input	Output
3 7 17 9 3 5 1	4

System of equations

$$\begin{cases} 7 & \equiv & 3 & (\mathrm{mod}\ 4) \\ 17 & \equiv & 5 & (\mathrm{mod}\ 4) \\ 9 & \equiv & 1 & (\mathrm{mod}\ 4) \end{cases}$$

is satisfied and it is easy to verify that it is not satisfied for $m > 4$.

Input	Output
3 4 6 5 2 2 2	1

System of equations

$$\begin{cases} 4 & \equiv & 2 & (\mathrm{mod}\ 1) \\ 6 & \equiv & 2 & (\mathrm{mod}\ 1) \\ 5 & \equiv & 2 & (\mathrm{mod}\ 1) \end{cases}$$

is satisfied and it is easy to verify that it is not satisfied for $m > 1$.