

Problem C. Computer Network

Input file: *standard input*
 Output file: *standard output*
 Time limit: 1 second
 Memory limit: 256 mebibytes

The additive-increase/multiplicative-decrease (AIMD) algorithm is a feedback control algorithm best known for its use in TCP congestion control. AIMD combines linear growth of the congestion window when there is no congestion with an exponential reduction when congestion is detected. Multiple flows using AIMD congestion control will eventually converge to an equal usage of a shared link. (from Wikipedia)

You are given two arrays of n integers: a and b . You can perform operations on the array a . In one operation, you can let a_i become $a_i + 1$ for all $1 \leq i \leq n$, or let a_i become $\lfloor \frac{a_i}{2} \rfloor$ for all $1 \leq i \leq n$.

Find the minimum number of operations that you have to perform to transform a into b , or determine that it is impossible.

Input

The first line contains an integer n ($1 \leq n \leq 10^6$).

The second line contains the integer array a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$).

The third line contains the integer array b_1, b_2, \dots, b_n ($0 \leq b_i \leq 10^9$).

Output

Print the minimum number of operations needed, or -1 if it's impossible to transform a into b .

Examples

<i>standard input</i>	<i>standard output</i>
5 1 2 3 4 5 6 6 6 6 7	9
3 2 3 4 1 2 3	-1
2 65536 65537 1 2	32