**Help Fredo**

Max. Marks: 100

Fredo is assigned a task today. He is given an array AA containing NN integers. His task is to update all elements of array to some minimum value xx, that is, A[i]=xA[i]=x ; 1≤i≤N1≤i≤N such that product of all elements of this new array is strictly greater than the product of all elements of the initial array. Note that xx should be as minimum as possible such that it meets the given condition. Help him find the value of xx.

**Input Format**:  
The first line consists of an integer NN , denoting the number of elements in the array.  
The next line consists of NN space separated integers, denoting the array elements.

**Output Format**:  
The only line of output consists of value of xx.

**Input Constraints**:  
1≤N≤1051≤N≤105  
1≤A[i]≤10101≤A[i]≤1010; A[i]A[i] denoting an array element.

**SAMPLE INPUT**

5

4 2 1 10 6

**SAMPLE OUTPUT**

4

**Explanation**

Initial array product =4∗2∗1∗10∗6=480=4∗2∗1∗10∗6=480  
If all elements become 44, then product =4∗4∗4∗4∗4=1024=4∗4∗4∗4∗4=1024. Had all elements become 33, product would be =243=243 which is less than 480480. Hence, value of xx is 44.

**Time Limit:**1.0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded if any testcase passes.

**Allowed Languages:**C, C++, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Scala 2.11.8, Swift, Visual Basic

<https://www.hackerearth.com/challenge/competitive/april-circuits-17/algorithm/help-fredo/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int n = int.Parse(Console.ReadLine());

double[] arr = Array.ConvertAll(Console.ReadLine().Trim().Split(' '), e => double.Parse(e));

//int[] arr = { 4, 2, 1, 10, 6 };

//long[] arr = { 2, 2, 2, 2, 2 };

//long[] arr = { 10000000000, 10000000000, 10000000000 };

//long[] arr = { 100, 100, 100 };

double prod = 1.0;

double len = arr.Length;

for (int i = 0; i < arr.Length; i++)

{

prod \*= Math.Pow( arr[i], 1.0/len);

}

if (arr.Distinct().Count() == 1)

{

Console.WriteLine(arr[0] + 1);

}

else

{

Console.WriteLine(Math.Ceiling(prod));

}

Console.ReadLine();

}

}

}