

E. Bear and Paradox

time limit per test: 3.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Limak is a big polar bear. He prepared n problems for an algorithmic contest. The i -th problem has **initial** score p_i . Also, testers said that it takes t_i minutes to solve the i -th problem. Problems aren't necessarily sorted by difficulty and maybe harder problems have smaller initial score but it's too late to change it — Limak has already announced initial scores for problems. Though it's still possible to adjust the speed of losing points, denoted by c in this statement.

Let T denote the total number of minutes needed to solve all problems (so, $T = t_1 + t_2 + \dots + t_n$). The contest will last exactly T minutes. So it's just enough to solve all problems.

Points given for solving a problem decrease linearly. Solving the i -th problem after x minutes gives exactly $p_i \cdot (1 - c \cdot \frac{x}{T})$ points, where $c \in [0, 1]$ is some real constant that Limak must choose.

Let's assume that c is fixed. During a contest a participant chooses some order in which he or she solves problems. There are $n!$ possible orders and each of them gives some total number of points, not necessarily integer. We say that an order is *optimal* if it gives the maximum number of points. In other words, the total number of points given by this order is greater or equal than the number of points given by any other order. It's obvious that there is at least one optimal order. However, there may be more than one optimal order.

Limak assumes that every participant will properly estimate t_i at the very beginning and will choose some optimal order. He also assumes that testers correctly predicted time needed to solve each problem.

For two distinct problems i and j such that $p_i < p_j$ Limak wouldn't be happy to see a participant with strictly more points for problem i than for problem j . He calls such a situation a *paradox*.

It's not hard to prove that there will be no paradox for $c = 0$. The situation may be worse for bigger c . What is the maximum real value c (remember that $c \in [0, 1]$) for which there is no paradox possible, that is, there will be no paradox for **any optimal order** of solving problems?

It can be proved that the answer (the maximum c as described) always exists.

Input

The first line contains one integer n ($2 \leq n \leq 150\,000$) — the number of problems.

The second line contains n integers p_1, p_2, \dots, p_n ($1 \leq p_i \leq 10^8$) — initial scores.

The third line contains n integers t_1, t_2, \dots, t_n ($1 \leq t_i \leq 10^8$) where t_i is the number of minutes needed to solve the i -th problem.

Output

Print one real value on a single line — the maximum value of c that $c \in [0, 1]$ and there is no optimal order with a paradox. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6} .

Namely: let's assume that your answer is a , and the answer of the jury is b . The checker program will consider your answer correct if $\frac{|a-b|}{\max(1,b)} \leq 10^{-6}$.

Examples

input

Copy

```
3
4 3 10
1 1 8
```

VK Cup 2016 - Round 1

Finished

Practice



→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Practice

You are registered for practice. You can solve problems unofficially. Results can be found in the contest status and in the bottom of standings.

→ Clone Contest to Mashup

You can clone this contest to a mashup.

Clone Contest

→ Submit?

Language: GNU G++11 5.1.0

Choose file: 选择文件 未选择任何文件

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

Submit

→ Last submissions

Submission	Time	Verdict
65730940	Nov/26/2019 08:03	Accepted

→ Problem tags

greedy math sortings *2800

No tag edit access

output Copy
0.6250000000

input Copy
4
7 20 15 10
7 20 15 10

output Copy
0.31901840491

input Copy
2
10 20
10 1

output Copy
1.0000000000

→ **Contest materials**

- Announcement (en) ×
- Tutorial (en) ×

Note

In the first sample, there are 3 problems. The first is (4, 1) (initial score is 4 and required time is 1 minute), the second problem is (3, 1) and the third one is (10, 8). The total time is $T = 1 + 1 + 8 = 10$.

Let's show that there is a paradox for $c = 0.7$. Solving problems in order 1, 2, 3 turns out to give the best total score, equal to the sum of:

1. solved 1 minute after the start: $4 \cdot (1 - 0.7 \cdot \frac{1}{10}) = 3.72$
2. solved 2 minutes after the start: $3 \cdot (1 - 0.7 \cdot \frac{2}{10}) = 2.58$
3. solved 10 minutes after the start: $10 \cdot (1 - 0.7 \cdot \frac{10}{10}) = 3$

So, this order gives $3.72 + 2.58 + 3 = 9.3$ points in total and this is the only optimal order (you can calculate total scores for other 5 possible orders too see that they are lower). You should check points for problems 1 and 3 to see a paradox. There is $4 < 10$ but $3.72 > 3$. It turns out that there is no paradox for $c = 0.625$ but there is a paradox for any bigger c .

In the second sample, all 24 orders are optimal.

In the third sample, even for $c = 1$ there is no paradox.

Supported by

