C++ Programming: Exam Variant 1 (Exam-2017-05-21)

Solutions for each task will be submitted in the form of compressed archive (.zip) files, containing .h and .cpp files.

Please be mindful of the strict input and output requirements for each task, as well as any additional requirements on running time, used memory, etc., as the tasks are evaluated automatically and not following the requirements strictly may result in your program's output being evaluated as incorrect, even if the program's logic is mostly correct.

For some of the tasks in this exam you are provided with files, which the Judge system places in your submitted solution. These files are the so-called "Solution Skeleton" and, depending on the task, may require you to write specific code for your solution to work (e.g. a Solution Skeleton may contain a file with the **`main()`** function defined, in which case your task will usually be to implement a class or function in another file, for the program to work correctly). DO NOT attempt to edit the Solution Skeleton files – the Judge system overwrites any files from the skeleton you submit, so it won't see your changes to them. Some tasks may contain additional files you can use (and edit) if you want – if so, this will be described explicitly in the task.

You can use C++03 and C++11 features in your code.

Unless explicitly stated, any integer **input** fits into **`int`** and any floating-point **input** can be stored in **`double`**.

Task 1 – Hot Water (E1-Task-1-Hot-Water)

The company "Water You Waiting For" provides hot water to houses in a village through a series of underground pipes, each house having its own pipe for hot water. But the pipes corrode over periods of years and need to be replaced. Since the company wants to save money, it only does checkups of the pipes once a year, always on the same date (April 1st). If a pipe needs to be replaced, it is replaced during the checkup.

But the company wants to save even more money by skipping checkups when they are not necessary. Since the pipes corrode at a constant speed, it is enough to have 2 measurements of a pipe's strength and calculate how much time remains before it needs to be replaced. This needs to be rounded-down to years, since the repair can only happen during a checkup – so if a pipe is going to break after 2 years and 11 months, the company needs to replace it after 2 years, because otherwise the pipe will be broken and leak water for 1 month until the 3rd year checkup.

Write a program which, given two arrays of consecutive strength measurements of all the pipes, calculates the years a pipe has remaining before it needs to be replaced (after the latest checkup).

Input

The first line of the standard input contains an array of positive integer numbers, separated by single spaces, representing the measurements of each pipe, made during last year's checkup – let's call them **`measurements1`**.

The second line of the standard input is analogous to the first, but contains the measurements from this year's checkup – let's call it **`measurements2`**.

So, the strength of pipe **`i`** last year was **`measurements1[i]`** and this year it is **`measurements2[i]`**.

Output

A single line, containing integers separated by single spaces, representing the years remaining until the corresponding pipe described in the input arrays must be replaced (counting from **`measurements2`**). If we call this array results, then pipe **`i`** has to be replaced **`results[i]`** years after **`measurements2[i]`** was measured.

Restrictions

`0 < measurements1[i] <= 1000000000`;

`0 <= measurements2[i] < measurements1[i]`;

`measurements1` and **`measurements2`** will have an equal, non-zero number of measurements, less than or equal to **`500`**.

The total running time of your program should be no more than **`0.1s`**

The total memory allowed for use by your program is **`5MB`**

Example I/O

Example Input	Expected Output	Explanation
5 4 3 3 2 2	1 1 2	Pipe 0 has suffered 5 – 3 = 2 damage – next year it will have 3 – 2 = 1 strength remaining and needs to be replaced, otherwise it will fail ~1.5 years from now. Pipe 1 will break exactly 1 year from now – i.e. replace during the checkup.
2 3 4 5 12 1 1 1 1 11	1 0 0 0 11	Pipe 0 and Pipe 4 get 1 damage per year, but the others lose more strength per year than they have remaining – we need to replace them now