Test Session for the Northwestern Europe Regional Contest 2024

NWERC 2024 Test Session

November 23, 2024

Problems

Alternative Encryption
Blackboard
Consolidating Windows

This page is intentionally left blank.

Alternative Encryption

Time limit: 1 second

The enigma, an encryption machine that solves this problem.
CC BY 2.0 by William Warby on Wikimedia Commons

In the fight against theft of intellectual property by rivalling universities, TU Delft has decided to implement a system of secure communication, to be called New Well-Encrypted Remote Communication. All internal communication will be encrypted before being sent over the network, and then decrypted upon arrival. They have already set up the infrastructure, they have come to you for the encryption.

You are tasked to design an algorithm for both encryption and decryption of text consisting of only English lowercase letters. You do not need to tell them how you do it, in fact, they encourage you to keep it a secret, to improve security. Your algorithm only needs to meet the following criteria:

- No matter what the text is, encrypting it and then decrypting the result should obviously yield the original text.
- To make sure there are no issues with the transmission, the encrypted text should also consist of only English lowercase letters and should have the same number of letters as the original text.
- To make sure the code cannot be broken easily, for all i, the ith letter of the encrypted text should differ from the ith letter of the original text.

The word “nwercjury” for example may not be encrypted as “irritating”, because the number of letters does not match. Nor may you encrypt it as “imbecilic”, since the fifth letter is a ‘c’ in both. An example of an acceptable encryption is “fantastic” (both have an ‘n’ and a ‘c’, but in different positions).

Your program will be run twice for each test case. In the first pass, your program will be given a number of strings to encrypt. In the second pass, your program will be given the strings as encrypted by the first pass, which it should then decrypt to retrieve the original input. Your submission may take up to 1 second for each pass.

A testing tool is provided to help you develop your solution.

Input

The input consists of:

- One line with either “encrypt” or “decrypt”, indicating the action your program has to perform.
- One line with an integer n ($1 \leq n \leq 1000$), the number of strings.
- n lines, each with a string s ($1 \leq |s| \leq 100$), the text to encrypt or decrypt.
  All input strings consist of only English lowercase letters (a-z).

Output

For each string, output its encryption or decryption, as required.

Sample Case 1

Sample Input **Pass 1** **Sample Output**

```
encrypt
3
plaintext
nwerc
correct
```

```
encrypted
delft
balloon
```

Sample Input **Pass 2** **Sample Output**

```
decrypt
3
encrypted
delft
balloon
```

```
plaintext
nwerc
correct
```

Blackboard

Time limit: 6 seconds

You find yourself in a room with a blackboard that has n positive integers written on it. You like it when things are organized, but this blackboard is one big mess: the numbers are all over the place, with a mix of very small and very large numbers.

To organize things, you will split the numbers into smaller numbers, one at a time, such that the total sum remains the same. Thus, in one operation, you can choose any value x from the blackboard, erase it, and replace it with two positive *real* numbers y and z such that $x = y + z$. Your goal is to ensure that the largest value on the blackboard is at most k percent larger than the smallest value.

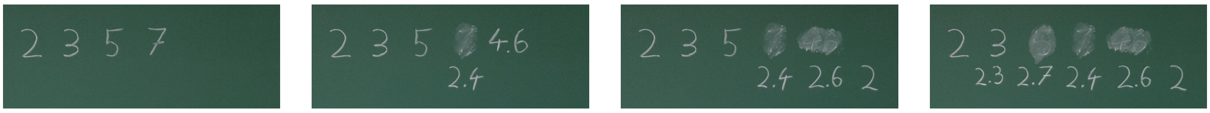

Figure 1: Illustration of Sample Input 1. The 7 can be replaced by 2.4 and 4.6. The 4.6 can in turn be replaced by 2.6 and 2. Finally, the 5 can be replaced by 2.3 and 2.7. After that, the largest value (3) is 50% larger than the smallest value (2).

Determine the minimum number of operations required to achieve this goal.

Input

The input consists of:

- One line with two integers n and k ($1 \le n \le 10\,000$, $0 \le k \le 100$), the initial number of integers on the blackboard and the required percentage of maximal difference.
- One line with n integers a ($1 \le a \le 10^9$), the initial integers on the blackboard.

Output

Output the minimum number of operations required to ensure that the largest value on the blackboard is at most k percent larger than the smallest value.

Sample Input 1	Sample Output 1

`4 50`
`2 3 5 7`
	`3`

Sample Input 2	Sample Output 2

`2 20`
`7 4`
	`1`

Consolidating Windows

Time limit: 1 second

Bricked-up windows. CC BY 3.0 by Roger Veringmeier

In the 19th century, windows were a costly luxury in Delft, as you had to pay tax for every single window in your home. Only the wealthy people could afford more than a few, while many poorer residents bricked up some of their windows to avoid the enormous costs introduced by the new policy.

Not wanting to sacrifice any of the precious sunlight in her home, Corry came up with an ingenious idea: since the tax was based solely on the number of windows and not their size, she decided to install fewer but larger windows instead of many small ones. In each room, Corry had two square windows which she planned to replace by one single, square window to halve the tax. The area of this new window should equal the combined area of the two smaller ones. Given the side lengths of the two smaller windows, what is the side length of the larger window?

Input

The input consists of:

- One line with two integers a and b ($1 \leq a, b \leq 10^9$), the side lengths of the two smaller square windows.

Output

Output the side length of a square window that has the same area as the two smaller windows combined.

Your answer should have an absolute or relative error of at most 10^{-8}.

Sample Input 1	Sample Output 1
3 4	5.0

Sample Input 2	Sample Output 2
10000000 320912	10005147.900543199866