Maratona SBC de Programação 2022

This problem set is used in simultaneous contests:
Maratona SBC de Programação SBC 2022
Tercera Fecha Gran Premio de México 2022
Gran Premio de Centroamérica 2022
Torneo Argentino de Programación 2022
The 2022 ICPC Bolivia Preliminary Contest

October 8th, 2022

Warmup session

General Information

This problem set contains 5 problems, some of them may have the same statement; pages are numbered from 1 to 6, without considering this page. Please, make sure that your book is complete.

A) Program name
1) Solutions written in C/C++ and Python, the filename of the source code is not significant, can be any name.
2) Solutions written in Java, filename should be: *problem_code*.`java` where *problem_code* is the uppercase letter that identifies the problem. Remember in Java the main class name and the filename must be the same.
3) Solutions written in Kotlin, filename should be: *problem_code*.`kt` where *problem_code* is the uppercase letter that identifies the problem. Remember in Kotlin the main class name and the filename must be the same.

B) Input
1) The input must be read from *standard input*.
2) The input is described using a number of lines that depends on the problem. No extra data appear in the input.
3) When a line of data contains several values, they are separated by *single* spaces. No other spaces appear in the input.
4) Every line, including the last one, ends with an end-of-line mark.
5) The end of the input matches the end of file.

C) Output
1) The output must be written to *standard output*.
2) When a line of results contains several values, they must be separated by *single* spaces. No other spaces should appear in the output.
3) Every line, including the last one, must end with an end-of-line.

Promotion:

Sociedade Brasileira de Computação

Problem A
Factorial

The *factorial* of a positive integer number N, denoted as $N!$, is defined as the product of all positive integer numbers smaller or equal to N. For example $4! = 4 \times 3 \times 2 \times 1 = 24$.

Given a positive integer number N, your task is to write a program to determine the smallest number k such that $N = a_1! + a_2! + \ldots + a_k!$, where every a_i, for $1 \leq i \leq k$, is a positive integer number.

For example, if $N = 10$ the answer is 3, because it is possible to write N as the sum of 3 factorial numbers: $10 = 3! + 2! + 2!$. If $N = 25$ the answer is 2, because it is possible to write N as the sum of 2 factorial numbers: $25 = 4! + 1!$.

Input

The input consists of a single line, which contains an integer N $(1 \leq N \leq 10^5)$.

Output

Your program must output a single line, containing an integer number, representing the smallest amount of factorial numbers which sum equals N.

Input example 1	Output example 1
10	3

Input example 2	Output example 2
25	2

Problem B
Forgotten Score

John is not very good at math but he learned that the average of two numbers is the sum of these two numbers divided by two. That is, the average of two numbers A and B is $M = \frac{A+B}{2}$.

The calculus teacher told John the scores he got on two tests he took. Both scores were integer numbers between 0 and 100. John immediately averaged the two tests, which also resulted in an integer number.

But John is very forgetful, and now he can't remember both scores he got on the tests. He can remember only one them. Luckily, he can also remember the average between the two scores.

Can you help John determine the score of the test he does not remember?

Input

The first line of input contains a single integer number A ($0 \le A \le 100$), representing the score of the test John remembers. The second line contains a single integer number M ($0 \le M \le 100$), representing the average between the two test scores.

Output

Your program must output a single line, containing an integer number, representing the score of the test which John can not remember.

Input example 1	Output example 1
100 70	40

Input example 2	Output example 2
80 75	70

Input example 3	Output example 3
1 50	99

Problem C

Lavaspar

Word Search is a well-known hobby, although it is losing some of its prestige in recent years. The goal in this game is to find words in an array, where each cell in this matrix contains a letter.
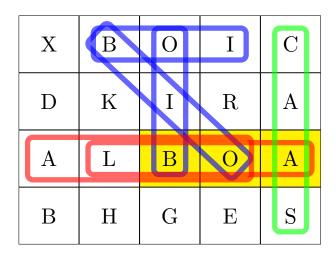
Bibika and her brother were playing Word Search, but soon they lost interest in the game, as finding all the words was getting relatively easy. Bibika would like to take her brother away from the computer, she searched the internet for games of the same style and ended up finding the *Lavaspar Hunting*.

Lavaspar Hunting is a game that follows the same idea of the famous Word Search. However, instead of simply having to find a word in the matrix, the goal is to find any anagram of the word, making the game more difficult and interesting. The anagram can be found in a row, column or diagonal.

An anagram is a word formed by rearranging the letters of another. Sometimes, an anagram does not exist as a word in the language, but this does not matter. BALO, LOBA and AOLB are examples of anagrams of the word BOLA.

Bibika realized that it was possible for the same cell in the matrix to make part of anagrams of different words and then she started to call these *special cells*.

Now she would like to know, given an array configuration and a collection of words, how many special cells are there?

The picture above illustrates the first example, where the collection of words consists of three words: BOLA, CASA and BOI. The rectangles of each color represent anagrams of different words from the entry. The 3 special cells are painted yellow.

Input

The first line contains two integers, L and C, which correspond to the number of lines and columns of the array, respectively.

Each one of the next L lines contains a word with C letters.

These lines are followed by a line containing an integer, N, which is the number of words in the collection of words to follow.

Finally, there are now N lines, each of which contains a word in the collection.

All characters in the array and in each word of the collection of words is a capital letter of the English alphabet.

No two of the N words in the collection are anagrams of each other.

- $2 \leq L, C \leq 40$.

- $2 \leq N \leq 20$.

- The number P of letters of each of the N words is in the interval $2 \leq P \leq \min(15, \max(L, C))$.

Output

The output consists of a single line that contains an integer corresponding to the number of special cells.

Input example 1	Output example 1
4 5 XBOIC DKIRA ALBOA BHGES 3 BOLA CASA BOI	3

Input example 2	Output example 2
3 3 AAB ABA BAA 2 ABA BBB	3

Input example 3	Output example 3
2 4 AAAA AAAA 2 AAA BBB	0

Problem D
Sticker Album

The Sticker Album of the ICPC 2021 Nlogonian Subregional just came out! Competitive programming hooligans all over the country are buying albums and collecting stickers, to celebrate the competition.

This sticker album is special because all stickers are equal: a picture of this year's trophy. To complete the album, you just need to collect enough stickers to fill all the slots in it.

You may be asking yourself: where is the fun in collecting those stickers? Well, to make things interesting, the stickers are sold in packets, each with a random number of stickers! Fans celebrate when they find a high number of stickers in a packet, make fun of those who got unlucky and found low numbers of stickers, and brag about filling their whole albums with just a few packets.

You just acquired your own album, and want to start filling it! But before buying your first sticker packets, you wondered: on average, how many packets does one need to buy in order to fill an album?

Input

The only input line contains three integers N, A and B, separated by a single space, satisfying $1 \leq N \leq 10^6$, $0 \leq A \leq B \leq 10^6$ and $B > 0$, where:

- N is the number of stickers it takes to fill an album;

- A is the minimum number of stickers in a packet;

- B is the maximum number of stickers in a packet.

The number of stickers in each packet is uniformly distributed in the closed interval $[A, B]$.

Output

The output consists of a single line, which must contain the expected number of packets it takes to complete an album. The number will be considered correct if it is within an absolute or relative error of 10^{-5} of the correct answer.

Input example 1	Output example 1
40 0 2	40.33333

Input example 2	Output example 2
100 1 10	18.72727

Input example 3	Output example 3
30 3 3	10.00000

Input example 4	Output example 4
314 5 8	48.74556

Problem E
Game Show

The Society of Bright Competitors (SBC) organizes television shows to its members (and currently it also broadcasts online!). SBC uses a a system of credits called *sbecs*, which can be used by players to participate in competitions or can be exchanged for prizes at the end of each season. SBC started a new type of game, and needs to do some simulations to avoid very large losses in the prize pool!

Ricardo is going to try the new game. He must bet 100 sbecs, which are transferred to his game balance. Then, a sequence of boxes is positioned. The game consists of rounds, and the maximum number of rounds is equal to the number of boxes. At each round, Ricardo decides whether to open the next box or to quit the game. If Ricardo quits, he gets the current balance of sbecs back. If Ricardo opens the next box, its content, which is a secret number, is added to his balance and the game continues. As the secret number in the box may be negative, Ricardo may end up at a loss! The game ends when Ricardo decides to quit or when the last box is opened.

SBC hired you to test the game. From the content of the boxes, you must decide what would be the largest possible balance that Ricardo could get.

Input

The first input line contains an integer C, $1 \leq C \leq 100$, which is the number of boxes in the game. After the first input line, there are C more lines. Each of the C lines contains the secret number of a box. The lines are in the same order of the boxes. The secret numbers are integers, V, $-1000 \leq V \leq 1000$.

Output

Output a line containing an integer which is the largest possible balance that Ricardo may get, given that sequence of boxes.

Input example 1	Output example 1
4 -1 -2 -3 -4	100

Input example 2	Output example 2
5 -10 20 -30 40 -50	120