

Computer Science I Program 0: Sign of Trouble (Strings)

Please Check Webcourses for the Due Date

Please read the whole assignment before you start coding

Objective

Give practice with reading and writing to standard input in C.

Give practice with loops and conditionals in C.

Give practice with strings in C.

Background Story

Your newest project is working on revitalizing the movie industry. There will be a lull in the upcoming movie productions due to all the strikes, and you want to support the area by renovating a local big box store into a theater.

The first major change is to fix the signage around the store to reflect the new purpose of the space. The letters themselves of the old store are fine, and we still want a sign in the same places. The message on the sign will change, but to save costs we will reuse as many letters as possible.

Problem

Given the original message and the new message determine how many new letters will need to be purchased to create the new sign.

Input

The first line of input will contain a sequence of characters (uppercase letters and spaces) denoting the original message of the sign. The second line of input will contain a sequence of characters (uppercase letters and spaces) denoting the desired resulting message of the sign.

You are guaranteed that each message will have at most 100,000 characters before the end of line characters.

Output

Output a single integer representing the number of letters that will need to be purchased.

Sample Input	Sample Output
ELECTRONICS SNACKS AND DRINKS	10
SOFA SINK AND SUCH REGRET CINEMAS	8

Sample Explanation

In the first sample case, we can use the letters 'C', 'R', 'N', 'I' and 'S' in making the second sign, leaving $6 + 3 + 6 - 5 = 10$ letters ('A', 'K', 'S', 'A', 'N', 'D', 'D', 'N', 'K' and 'S') that need to be purchased. Notice that there is only 1 'S' in the first sign but 3 S's in the second sign, so we have to purchase 2 S's. Alternatively, if we have equal or more of a letter in the old sign, then we do not need to purchase any of that letter.

In the second sample case, we can use 'I', 'A', 'N', 'S' and 'C' from the first sign leaving us $6 + 7 - 5 = 8$ letters left to buy: 'R', 'E', 'G', 'R', 'E', 'T', 'E' and 'M'.

Hints

Please use the fgets function to read in both strings. Make sure you allocate enough memory for both strings (100,002 chars – extra char for newline character and the null character). (You can play around with alternative functions later but if everyone just follows this directive students' grades will be higher and my TAs will be able to grade the program faster!)

Please make sure to use a frequency array. The details of this will be briefly discussed in lecture, so make sure you attend class and pay attention. In short in a frequency array for characters, index i will store the number of occurrences of the letter i , where $i=0$ corresponds to A, $i=1$ corresponds to B and so forth. If c is a char variable storing an uppercase letter, then the expression $c - 'A'$ equals the 0 to 25 equivalent of the letter c is storing, since Ascii values of uppercase characters are sequential.

Please assume that your program will be tested only on cases where the input format is followed. Namely, after reading in each string, do NOT check if the characters in it are only uppercase letters and spaces.

Implementation Requirements/Run Time Requirements

1. Please use either the malloc or calloc functions to dynamically allocate memory for the two input strings, since they can be quite large. (It's possible that if these arrays are statically allocated, on some systems a program would not correctly run.)
2. For full credit, your algorithm should run in linear time in the length of the input plus the alphabet size. (This means the total number of simple operations should be no more than a constant multiple of $n+26$, where n represents the sum of lengths of the two input strings.) This means that after you read through the input, you should have a fixed number of simple loops that example each letter in the input strings and each location in the two frequency arrays you create.

Deliverables

1. Please submit a single source file, `signoftrouble.c`, via Webcourses.