Sign of Trouble

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.

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 of 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

Explanation

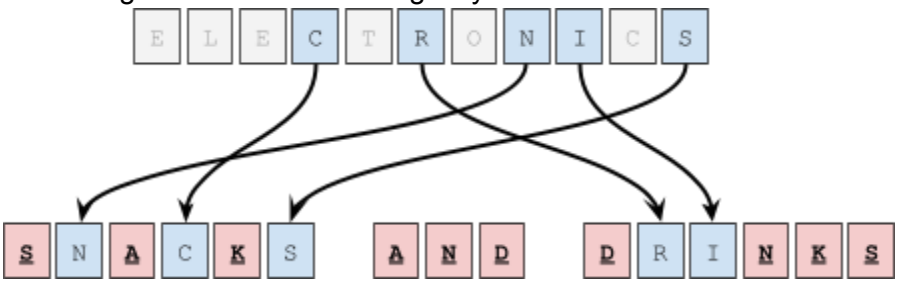
Case 1
The original sign is:

ELECTRONICS

The desired sign is:

SNACKS AND DRINKS

We can use the existing letters in the following way.



The letters in red (bolded and underlined) are the ones we will need to purchase. The letters in blue are the ones that can be reused. It should be noted that there are other ways the letters could have been assigned; the way suggested is one of the optimal ways.

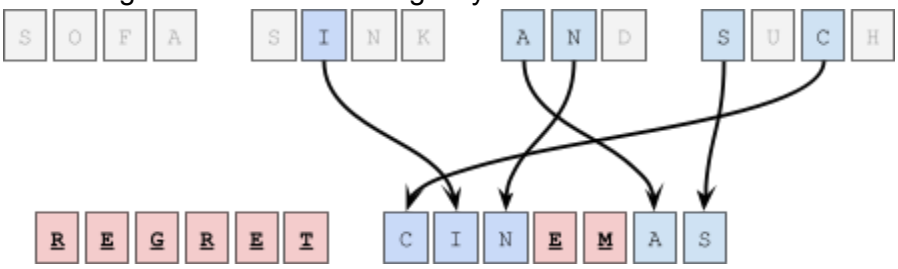
Case 2
The original sign is:

SOFA SINK AND SUCH

The desired sign is:

REGRET CINEMAS

We can use the existing letters in the following way.



The letters in red (bolded and underlined) are the ones we will need to purchase. The letters in blue are the ones that can be reused. Again, there are other ways the letters could have been assigned; the way suggested is one of the optimal ways.

Hints

Data Sanitization: In your intro to C class your teacher may have forced you to ensure that the content you are reading is correctly formatted. For this class the input can always assume to be within the input specification. In other words you do NOT need to sanitize the data.

Reading Input: I recommend reading using one of the following functions,

- fgets
- getchar
- getc

Notes on fgets: You need your buffer to have enough room for a null terminator and the new line characters in addition to all the other characters you will read in. This means you will need at least 2 “extra spots” in your array if you are reading in the line into a string using fgets.

Dynamic Memory: If you are creating a string of a “large” size (around 100k), then you may need to allocate your string dynamically. The memory you create statically is limited in C by some system parameters. However, you should have no issue creating the same memory on the same system using malloc or calloc.

Output: You should print out your answer after reading in the lines. You don’t need to print the output beside the input. That would be weird.

Independent Runs: Your program should only read in one message pair. Your program should stop after outputting the answer. If your program does not stop after reading in the messages your program will be given no credit due to failing to execute in the allotted time. Your program should not try to use the letters from a different message to complete the desired message. That is the “ELECTRONICS” message **cannot** be used to complete the “REGRET CINEMAS” message from the given samples.

I BETTER NOT FIND THIS ON THE INTERNET (OUTSIDE OF WEBCOURSES)!!!

Grading Criteria

- Read/Write from/to **standard** input/output (e.g. scanf/printf and no FILE *)
    - 10 points
- Good comments, whitespace, and variable names
    - 15 points
- No extra input output (e.g. input prompts, “Please enter the original sign’s message:”)
    - 10 points
- Read in all the input
    - 5 points
- Loop over all characters in both messages
    - 10 points
- Programs will be tested on 10 cases
    - 5 points each

No points will be awarded to programs that do not compile using “gcc -std=gnu11 -lm”.

Sometimes a requested technique will be given, and solutions without the requested technique will have their maximum points total reduced. For this problem use a loop and conditional. ***Without this programs will earn at most 50 points!***

Any case that causes a program to return a non-zero return code will be treated as wrong. Additionally, any case that takes longer than the maximum allowed time (the max of {5 times my solutions time, 10 seconds}) will also be treated as wrong.

No partial credit will be awarded for an incorrect case.