

# Hidden Message

## Step One

The program is passed two messages in the form of strings. These messages have been converted to Morse code tokens. The goal is to find all of the possible unique sequences of remaining tokens after removing the second message's tokens from the first message's tokens. There are 3 different types of tokens in the Morse code message.

- Dot (\*)
- Dash (-)
- Blank (\_)

Every letter in the message is separated by a single blank character (\_) and every word is separated by 3 blank characters (\_\_\_).

### Example:

- Given: AB  
○ \* \_ \_ \* \* \*
- Remove: R  
○ \* \_ \*
- There are 6 different ways to remove R from AB, we will call these deletion paths.  
○ X X \_ - X \* \*  
○ X X \_ - \* X \*  
○ X X \_ - \* \* X  
○ X - \_ X X \* \*  
○ X - \_ X \* X \*  
○ X - \_ X \* \* X
- But there are only 2 possible unique sequences of remaining tokens within the set of deletion paths.  
○ \_ - \* \*  
○ - \_ \* \*
- So the final result for this example would be 2.

Write a program that can calculate the number of possible unique sequences of remaining tokens for removing one Morse code message from another. This program should be able to solve the example below in under 10 seconds.

- Given: Hello World  
○ \* \* \* \* \_ \* \_ \* \_ \* \* \_ \* \_ \* \_ \_ \_ \* \_ \_ \_ \_ \* \_ \* \_ \* \_ \* \*
- Remove: Help  
○ \* \* \* \* \_ \* \_ \* \_ \* \_ \* \_ \_ \*

- ANSWER: 1311

## Step two

Using the set of possible unique sequences of remaining tokens from step one, find the possible unique sequences of remaining tokens where you must remove a second set of tokens from each of the sequences. All remaining tokens would be kept in the same order when removing the second sequence of tokens. Return all of the possible, unique sequences of remaining tokens after removing both messages.

### Example:

- Given: ABCD
  - \* \_ \_ \* \* \* \_ \_ \* \_ \_ \* \*
- Remove: ST
  - \* \* \* \_ \_
- Then Remove: ZN
  - \_ \_ \* \* \_ \*
- One solution path would look like:
  - Start:
    - \* \_ \_ \_ \* \* \* \_ \_ \* \_ \_ \* \*
  - Remove ST:
    - x \_ \_ \_ x x \* x x \* \_ \_ \* \*
  - Then Remove ZN:
    - x x \_ x x x x x x x - \* x x x \*
  - The sequence of remaining characters
    - \_ \_ \* \*
- There are 5 sequences of remaining characters for this example:
  - \_ \* \*
  - \_ \* \_ \*
  - \_ \_ \* \*
  - \* \_ \_ \*
  - \* \_ \_ \*

Write a program that will find all of the possible unique sequences of remaining characters after removing 2 Morse code messages from an original message. This program should be able to calculate all of the sequences in the example below in less than 60 seconds and return the total number of distinct and valid sequences found.

- Given: The Star Wars Saga
  - \_ \_ \* \* \* \* \_ \_ \* \* \* \_ \_ \* \_ \* \_ \_ \* \_ \_ \* \* \* \_ \_ \* \* \_ \_ \* \_ \_ \*
- Remove: Yoda
  - \_ \* \_ \_ \_ \_ \* \* \_ \*
- And Remove: Leia

- \*\_\*\*\_\*\*\*\*\_  
• Expected Answer: 11474

## Program Specifications

- Input: Three command-line arguments denoting the original message, the first hidden message, and the second hidden message. The three messages will be in Morse code using the representation described here.
- Expected Output: Total number of distinct and valid remaining token sequences in the original message.
- Bounds: Original Message less than 100 Morse code characters.