



# Practice Arena

Practice problems aimed to improve your coding skills.

- 📁 PRACTICE-02\_SCAN-PRINT
- 📁 PRACTICE-03\_TYPES
- 📁 LAB-PRAC-02\_SCAN-PRINT
- 📁 LAB-PRAC-01
- 📁 PRACTICE-04\_COND
- 📁 BONUS-PRAC-02
- 📁 LAB-PRAC-03\_TYPES
- 📁 PRACTICE-05\_COND-LOOPS
- 📁 LAB-PRAC-04\_COND
- 📁 LAB-PRAC-05\_CONDDOOPS
- 📁 PRACTICE-07\_LOOPS-ARR
- 📁 LAB-PRAC-06\_LOOPS
- 📁 LAB-PRAC-07\_LOOPS-ARR
- 📁 LABEXAM-PRAC-01\_MIDSEM
- 📁 PRACTICE-09\_PTR-MAT
- 📁 LAB-PRAC-08\_ARR-STR
  - ❓ Il fratello di Fibonacci
  - ❓ Hidden Palindrome
  - ❓ Hush Hush Hash
  - ❓ Maximum Match
  - ❓ El secreto de sus l
  - ❓ Star Replacement
  - ❓ Stronger together
  - ❓ Rigorous and repeated redaction
  - ❓ strnrev
  - ❓ Monster and Mini Multiply
  - ❓ Clash of the Substrings
  - ❓ Personalizing Emails
- 📁 PRACTICE-10\_MAT-FUN
- 📁 LAB-PRAC-09\_PTR-MAT
- 📁 LAB-PRAC-10\_MAT-FUN
- 📁 PRACTICE-11\_FUN-PTR
- 📁 LAB-PRAC-11\_FUN-PTR
- 📁 LAB-PRAC-12\_FUN-STRUC
- 📁 LABEXAM-PRAC-02\_ENDSEM
- 📁 LAB-PRAC-13\_STRUC-NUM
- 📁 LAB-PRAC-14\_SORT-MISC

# Rigorous and repeated redaction

## LAB-PRAC-08\_ARR-STR

**Rigorous and repeated redaction [20 marks]**

---

**Problem Statement**

You will be given a non-empty string consisting only of upper case and lower case English alphabets in the first line of the input. The string will contain no more than 999 characters. The second line of the input will contain another non-empty string of only upper and lower case English alphabets with no more than 99 characters. The first string is a message you wish to publish online whereas the second string is sensitive material (e.g. passwords etc). You have to replace all instances of occurrences of the sensitive string in the original string with the letters "XXX" (without quotes). This process is often called *redaction*.

In the first line of your output, print how many times does the first letter of the message appear in the message either in lower or upper case. In the second line, print the redacted string. If the sensitive material is not present in the message at all, just print the original message in the second line.

**Caution**

1. Be careful about extra/missing lines and extra/missing spaces in your output.
2. Although while counting how many times the first character of the message appears in the message, you have to count both upper and lower case occurrences, when checking for occurrences of the sensitive string, do so in a case sensitive manner (see example below).

**HINTS:**

1. The string.h library provides the strlen function to calculate the length of a string and the strstr function to find the occurrence of substrings in a given string. However, you need to be able to manipulate pointers to use strstr.
  2. This question can be solved without using pointers as well.
- 

**EXAMPLE:****INPUT**

hellohitHisismehighupinthesky  
hi

**OUTPUT:**

6  
helloXXXtHisismeXXXghupinthesky

---

**Grading Scheme:**

Total marks: **[20 Points]**

There will be partial grading in this question. There are two lines in your output. Printing each line correctly, in the correct order, carries some weightage. The first line carries 25% weightage and the

second line carries 75% weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

 **Start Solving!** (</editor/practice/6170>)