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_CONDLOOPS
- 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
- 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
    - Meanie Numbers
    - Rotate Then Rotate Code
    - The enigma that was Enigma
    - Save the Date
    - Pretty Patterns
    - Trivial Tic-Tac-Toe
    - How Mr C reads your code
    - Malloc Mystery
- LAB-PRAC-13_STRUC-NUM
- LAB-PRAC-14_SORT-MISC

Rotate Then Rotate Code

LABEXAM-PRAC-02_ENDSEM

Rotate Then Rotate Code [40 marks]

Problem Statement
Mr C had written a few messages to his friends and kept them on the desk. But his mischievous clone came and jumbled up the messages. Help Mr C figure out the original message. In your input you will be given two strings containing only lower-case English alphabets (no spaces, punctuation marks). Each string will be given to you on a separate line so you can use the gets() function to read the two strings. We promise you that both strings will have 99 characters or less.

Mr C suspects that the second string was obtained by taking the first string and performing two rotation operations on the first string. The operations are described below

1. First, the English alphabet was rotated to the right by k letters (k is not known to you). For example if k = 2, a would become c, b would become d, y would become a, z would become b, and so on. The clone first used this rotated alphabet to rewrite the original string. For example, if the original string was abc and k = 2 then the string now becomes cde
2. The new string is itself rotated to the right by l locations (l is not known to you). For example, if l = 2, the new string cde, rotated one location becomes ecd and rotated two locations becomes dec.

If the second string can be obtained from the first string by performing the above two rotations, you have to print in your output the word "YES" (without quotes) followed by a space followed by the value of k followed by a space followed by the value of l. If there is no way to obtain the second string from the first string using the rotations described above, print "NO" (without quotes) in your output.

Compulsory function usage in your code
In your code, you should write a function in the following format that takes in two strings and sets the variables isSame, k and l (sent as references) to their appropriate values if the first string can be converted to the second string. Be warned that not using such a function to write your code will cause you to lose a small number of manual grading marks.

```
void compare(char *str1, char* str2, int *k, int *l, int *isSame){
   // Write your code here
   // If str1 can be converted to str2, you can set *isSame = 1, else *isSame = 0
   // Similarly use k and l as well.
}

char str1[100], str2[100];
int k, l, isSame;
compare(str1, str2, &k, &l, &isSame);
```

Problem-specific Words of Caution:

1. **Do not forget to submit your code**. You can submit multiple times. Your last submission will get graded.
2. The number k must be between 0 and 25 (since k = 26 is the same as k = 0). Similarly, the number l must be between 0 and len - 1 where len is the length of the first string since rotating the string by len locations gives the string back.
3. Be careful that there is only one line in your output.
4. Do not worry, we will never give you a test case where two or more possible values of k and l exist which convert the first string to the second string. There will either exist no rotations or

else a single unique way of performing the two rotations so that the first string gets converted to the second string.

General Grading Policy

1. **TOTAL MARKS OF THE EXAM** 20 + 40 + 40 + 70 = 170
2. **TOTAL DURATION OF THE EXAM** 3 hours 30 minutes
3. See below for question-specific details of how partial marking would be done by the autograder in this question
4. Your submissions will be inspected by the autograder as well as a human grader
5. Human graders will (among other things) allot marks for the following

    1. Neatly structured code that uses at least one function other than the main function to process the input. The questions will usually suggest how to use functions to process the input. Submissions that ignore these suggestions and use only the main function to solve the entire problem, will lose a small fraction of marks.

    2. Proper and meaningful variable names

    3. Nice looking and consistent indentation

    4. At least a couple of comments explaining to the human grader what are you doing, especially when the steps are not obvious

    5. Comments, good indentation and meaningful variable names are very important for the human grader to understand what are you doing and why. If they cannot understand your code, do not expect them to give you (partial) marks either.

6. Solutions that indulge in hard-coding **will get a straight zero** even if they are passing some test cases. Hard-coding is a form of cheating strategy where someone write code of the form "if(input == A) printf(B)" without doing any calculations on A to obtain B. The values of A and B are either read from the evaluation/submission window or else guessed.
7. Be careful about extra/missing lines and extra/missing spaces if you do not want to lose autograder marks
8. Proportion of marks allotted to autograder (in particular, weightage to visible and hidden test cases) and human grader will be revealed when marks and grading rubrics are released
9. You are allowed to use the libraries stdio.h, math.h, string.h, stdlib.h **but not any other library**. Use of unpermitted libraries will carry a penalty. You may use any programming tools that we have discussed in lectures/tutorials or in lab questions such as arrays (1D, 2D, 3D, arrays of arrays etc), strings, loops, structures, functions, recursion, pointers, linked lists, stacks, queues, graphs, enumerations, flags, conditionals, global, static and shadowed variables.

EXAMPLE 1:
INPUT

abc
cdb

OUTPUT:
YES 1 2

Explanation: We first shift the alphabet by 1 position to the right i.e. a->b, b->c, c->d resulting in bcd. Then we rotate the string bcd two locations to the right to get cdb

EXAMPLE 2:
INPUT
abc
abc

OUTPUT:
YES 0 0

Explanation: No need to rotate the alphabet or else the string

----

Grading Scheme:
Total marks: **[40 Points]**

There will be no partial grading in this question. An exact match will receive full marks whereas an incomplete match will receive 0 points. Please be careful of missing/extra spaces and missing/lines (take help of visible test cases). There are 4 visible and 4 hidden test cases.

Start Solving! (/editor/practice/6246)