Lab 8
Strings - Structs

1 Lab Objectives

- Learning about Strings.
- Learning about Structures.
- More practice on loops and Arrays.
- Verify your solutions on hackerrank https://www.hackerrank.com/programming-lab-8-f2019.

2 Problem 1 - String Occurrence

You should write a program that takes two inputs:

1. A string line (use gets()/fgets() function to read a whole line).
2. A string word.

It should find **ALL** occurrences of the word in the input line. Your program should print the beginning index of each occurrence and it should print the total number of occurrences.
For example, if the input line is "aabacd aabaaba aabcaaba", and the input word is "aaba", your program should print:

```
Occurrence at index 0
Occurrence at index 7
Occurrence at index 10
Occurrence at index 19
Total number of occurrences = 4
```

Note:
You should think about special cases and handle them. Your program must NOT crash under any input.

3 Problem 2 - Count the Words

In this problem, you are given a line of text (use gets()fgets() to read a string line). Your task is to count the number of words in the input text. The input text will contain letters{a-z, A-Z, 0-9}, spaces and tabs ('\t').

Sample Input and Output:

- Input: "Hello World!" → Output: "2 words".
- Input: " Can you   count   the number of words" → Output: "7 words"
- Input: "   Be ware of specail   cases   " → Output: "5 words"

4 Problem 3 - Rational Numbers

A rational number consists of a *numerator* and a *denominator*. You should write a program that supports the following operations on rational numbers:

1. Addition
2. Subtraction
3. Multiplication
4. Division

In this problem, you're required to take as input two rational numbers, and the required operation. The output should print the expression being evaluated along with its result.

Notes:

- Each rational number should be in the reduced form (i.e: divide both numerator and denominator by their greatest common divisor (GCD))
- You are required to write a *function* for each of the above five operations.
- You are required to use structures to make a new data type called "Fraction", which contains 2 integers: numerator and denominator. You can read it with "%d/%d"
- The four functions should have the following prototypes:
  - *Fraction add(Fraction num1, Fraction num2);*
  - *Fraction subtract(Fraction num1, Fraction num2);*
  - *Fraction multiply(Fraction num1, Fraction num2);*
  - *Fraction divide(Fraction num1, Fraction num2);*

5 Problem 4 - Time

Write the following function assuming that you have a *time* structure that contains three members: *hours*, *minutes* and *seconds* (all of type *int*).
time split_time(long long total_seconds);
total_seconds is a time represented as the number of seconds since midnight. The function should return a structure containing the equivalent time in hours (0-23), minutes (0-59), and seconds (0-59).

6 Problem 5 - Circular Right Shift Subarray

Given array A of size n, we say that elements of A[i.....j] form a subarray. Your task is to perform circular right shift queries on subarrays of A . A circular right shift can be performed by reversing the array multiple times.
If we want to shif subarray A[i....j] to the right by k places, we can do that by calling a function that reverses a subarray.
Input Format :
first line contains n, size of the array
second line contains n integers of the array
third line contains q, number of queries
the next q lines each of them contains three integers
l, left index of the subarray
r, right index of the subarray
k, the number of elements to shift
Constraints :
$1 \leq n \leq 1000000$
$1 \leq q \leq 1000$
$1 \leq r - l + 1 \leq 1000$
k is non negative integer
no constraints on values of the array
Output Format :
print the array after performing all the queires
Sample Input :
5
1 2 3 4 5
2
0 1 1
2 4 10
Sample Output :
2 1 5 3 4

Explanation :
first query shifts subarray A[0..1] to the right by by 1
second query shifts subarray A[2...4] to the right by 10

7 Notes

- Read the lab **carefully** and stick exactly to what's required!
- Start in the lab early and come forward with your questions.
- You are required to bring your C programs for all problems to the lab on a CD / Flash memory / Laptop.
- Cheating will be severely penalized (for both parties). So, it is better to deliver nothing than deliver a copy!
- You are encouraged to ask any questions on Piazza, or in person.

Good Luck isA :)