

Introduction to Programming and Data Structures, 2023-24, Semester-II Assignment 03

Maximum Marks: 150

Topic: Structures and strings

Submission Deadline: **2023-Sep-21**

Clarification Deadline: **2023-Sep-15**

Here are 3 problems. You are to write C programs for the following problems. In each solution, you should take/give input/output only from/to a file only.

AP0301: Create a C program that implements rational number operations using structures.

Define a structure to represent rational numbers. Each rational number should have two components: numerator and denominator.

Implement the following rational number operations as separate functions:

- Addition of two rational numbers
- Subtraction of two rational numbers
- Multiplication of two rational numbers
- Division of two rational numbers

Implement a function to display rational numbers in the standard format (numerator/denominator).

In the main function, demonstrate the use of these operations by performing the following tasks:

- Create two rational numbers by reading their numerator and denominator from the user.
- Perform addition, subtraction, multiplication, and division of the two rational numbers using the implemented functions.
- Display the results of each operation in the standard format.

Ensure that your program handles division by zero (i.e., denominator should not be zero) and reduce fractions to their simplest form (e.g., 2/4 should be displayed as 1/2).

Provide meaningful comments and proper code documentation for better understanding.

Sample Input: from a file with filename “input_0301.txt”

First line contains number of test cases followed by the test cases. Each test case is of the form $a\ op\ c\ d$ where $op \in \{+, -, *, /\}$, a, b, c and d are *ints*.

AP0302: Create a C program that implements complex number operations using structures.

Define a structure to represent complex numbers. Each complex number should have two components: real and imaginary.

Implement the following complex number operations as separate functions:

- Addition of two complex numbers
- Subtraction of two complex numbers
- Multiplication of two complex numbers
- Division of two complex numbers

Implement a function to display complex numbers in the standard format ($a + bi$), where 'a' represents the real part, 'b' represents the imaginary part, and 'i' represents the imaginary unit.

In the main function, demonstrate the use of these operations by performing the following tasks:

- Create two complex numbers by reading their real and imaginary parts from the user.
- Perform addition, subtraction, multiplication, and division of the two complex numbers using the implemented functions.
- Display the results of each operation in the standard format.

Ensure that your program handles division by zero (i.e., denominator should not be zero).

Provide meaningful comments and proper code documentation for better understanding.

Sample Input: from a file with filename “input_0302.txt”

First line contains number of test cases followed by the test cases. each test case is of the form $a b op c d$ where $op \in \{+, -, *, /\}$, a, b, c and d are floats.

AP0303: • Assume place value of characters $\{a, b, \dots, z\}$ be $\{1, \dots, 25, 26\}$. Let us define the following operations to interchange them.

- Let `char_to_pos(.)` be a function that returns the positional value of a character.
E.g., $16 \leftarrow \text{char_to_pos}(p)$.
- Similarly, let `pos_to_char(.)` be a function that returns character value of a position.
E.g., $p \leftarrow \text{pos_to_char}(16)$

• Let S_1 and S_2 be two strings of lengths l_1 and $l_2 (\leq l_1)$ respectively. So given two strings, assign the longer string as S_1 and the shorter one as S_2 . Given length l_1 , the string is stored in $S_1[0]$ to $S_1[l_1 - 1]$. Let $S_a = S_1 + S_2$ and $S_m = S_1 * S_2$ where we define the operations ‘+’, ‘*’ over strings as follows.

.+ . \rightarrow if $S_a = S_1 + S_2$ then,

$$S_a[i] = S_1[i], \text{ if } l_2 \leq i \leq l_1 - 1$$

$$= \text{pos_to_char}((\text{char_to_pos}(S_1[i]) + \text{char_to_pos}(S_2[i])) \% 26), \text{ if } 0 \leq i \leq l_2 - 1$$

.* . \rightarrow if $S_m = S_1.S_2$ then,

$$S_m[i] = S_1[i], \text{ if } l_2 \leq i \leq l_1 - 1$$

$$= \text{pos_to_char}((\text{char_to_pos}(S_1[i]) * \text{char_to_pos}(S_2[i])) \% 26), \text{ if } 0 \leq i \leq l_2 - 1$$

• Finally, you are given two 2d-matrices in two files `input_0303.a.txt` and `input_0303.b.txt`. The first line of each file contains dimensions of the matrices and thereafter each row of the matrices. All strings in a row are space separated and contain smallcase characters only and have maximum length 15.

• Your task to find the multiplication of the two matrices (usual multiplication) which is also a matrix and output the matrix in a file `output_0303.txt` of which first line contains the dimension. If matrix multiplication is not possible, then write `0 0` in the first line only. In the end, free the memory occupied by the matrices if needed.

[35+35+100]

Note: For bad indentation of wrong filename, 10% marks will be automatically deducted.