

Systems Software
Assignment 1: Basic C programming

Assignment Objectives

Students should demonstrate the following abilities:

1. Use the **C struct** construct to define new data type
2. Use pointers to define input/output parameters in C functions
3. Write a complete C program that includes structures and functions to manipulate pointers

Assignment

You are working problems in which you must display your results as integer ratios; therefore, you need to be able to perform computations with common fractions and get results that are common fractions in reduced form. You want to write a program that will allow you to add, subtract, multiply, and divide several pairs of common fractions.

Define a C structure named **fraction** with two data members of type integer (**numerator** and **denominator**) as given below:

```
typedef struct{
    int numerator;
    int denominator;
} fraction;
```

Define the following functions:

add_fractions and **multiply_fractions** – Both methods accept two input parameters of type **fraction** and return the sum or the product of the input fractions through one output parameter of type **fraction**. Subtraction of two fractions is performed by changing the sign of the numerator of the second fraction and calling **add_fractions**. Similarly, multiplication of two fractions is performed by inverting the second fraction and calling **multiply_fractions**.

get_fraction – prompts the user for two integers and returns a structure of type **fraction**. The denominator should not be negative or equal to zero.

print_fraction - accepts an input parameter of type **fraction** and prints the fraction in the form numerator/denominator.

find_gcd, - accepts two input parameters of type **int** and returns their greatest common divisor (GCD) as an **int**.

reduce_fraction – accepts one input/output parameter of type **fraction** and reduces the input fraction using the GCD of the numerator and the denominator.

invert_fraction – accepts one input/output parameter of type **fraction** and inverts it (swap the numerator and the denominator).

Write a main method that interacts with the user to get two fractions and an operation and prints the result of the operation as a reduced fraction. Your program should keep asking the user if she/he wants to perform another operation until the user enters 'n'. You may add more functions in your program if you want but the functions above are required.

A sample run of the program is provided below.

Enter a common fraction as two integers separated by a slash> 1/4

Enter a common fraction as two integers separated by a slash> 3/4

Select an operation (+, -, *, /)> +

1/4 + 3/4 = 1

Do you want to perform another operation? (y/n)> y

Enter a common fraction as two integers separated by a slash> 2/3

Enter a common fraction as two integers separated by a slash> 1/-2

Invalid input – denominator must be positive.

Enter a common fraction as two integers separated by a slash> -1/2

Select an operation (+, -, *, /)> *

2/3 * -1/2 = -1/3

Do you want to perform another operation? (y/n)> y

Enter a common fraction as two integers separated by a slash> 1/2

Enter a common fraction as two integers separated by a slash> 3/2

Select an operation (+, -, *, /)> ?

Invalid operation – must be (+,-,*, or /)

Select an operation (+, -, *, /)> /

$$1/2 / 2/3 = 1/3$$

Do you want to perform another operation? (y/n)> y

Enter a common fraction as two integers separated by a slash> 9/4

Enter a common fraction as two integers separated by a slash> 5/2

Select an operation (+, -, *, /)> -

$$9/4 - 5/2 = -1/4$$

Do you want to perform another operation? (y/n)> n

Save your C program in a file named **fraction.c** and test it for different fractions and operations. Submit **fraction.c** on courseSite.