

Assignment #2

Exercise 1

Name: Sahand Sabour

Student ID Number: 1614650

Module code: EEE102

Problem Statement

The first problem (Exercise 1) asks us to design an iFraction class, which is derived as a sub-class to the already created Fraction. Essentially, all the characteristics that were defined for the fraction class should also be available for the iFraction class. Therefore, the mentioned characteristics would be overwritten in order for them to be usable by the iFraction objects. It is also mentioned that these mixed fractions (iFractions) are defined by the user and therefore, we must get an input from the user and consequently, design our program in a way that would be able to create an iFraction object from the given input.

Analysis

The input for this program would be an integer called choice which allows the user to travel around the user interface and use all the functionalities provided in the menu. Additionally, there is an integer called whichType, which would allow the user to choose between entering improper and mixed fractions. Moreover, there are integers for determining the decimal part, numerator and denominator of the mixed fraction as well as a string for when the user wants to convert a decimal to a fraction.

The output for this program can either be an iFraction, Fraction, or decimal depending on what options the user chooses to use.

There are several variables required in order to insure the functionality of this program. The main ones are the three regarding the decimal part, numerator and denominator of the mixed fraction. The variables that were mentioned in creating the Fraction class, such as greatest common integer, choice, etc., are also implemented here likewise.

Design

As the first step of the algorithm, a menu is designed to provide the user different options to use within the program. This menu consists of 9 different options that the user can choose from. Since this is a sub-class of the Fraction class, a lot of the functions that are used here work the same as the functions used for Fraction class and are only overloaded in order to be applicable to iFraction class objects.

In addition to the three main functions that were used in almost all of the parts of algorithm when designing the Fraction class (gcd finder, showFraction, normalise), the two functions convertF and convertIF were also created in order to convert Fraction objects to iFraction objects vice versa respectively.

convertF function:

Here, first we receive a Fraction object as an input to the function. Then we normalise this Fraction in order to ensure that it is in its simplest form. Next, based on the sign and magnitude of the numerator and denominator of the given fraction, we have a set of if conditions that would determine the equivalent mixed fraction (iFraction object).

If the absolute value of the numerator is smaller than the denominator, then the decimal part is zero and the numerator and the denominator of the mixed fraction are equal to the numerator and denominator of the improper fraction.

If the absolute value of the numerator is equal to the denominator, then the decimal part would be one and the numerator and the denominator would be zero and one respectively (To show that the fraction part of the mixed fraction is zero and we only have decimal part).

If the absolute value of the numerator is greater than the denominator, then we have three possibilities. If the numerator is negative, then the decimal part is minus the decimal part of the division of the absolute value of numerator divided by the denominator and the numerator is the remainder of this division while the denominator stays the same. If the numerator is zero, the decimal part would also be zero and the denominator would be set to one to present the simplest form. If the numerator is positive, the decimal part would be the decimal part of the numerator divided by the denominator, the numerator would be the remainder of the mentioned division while we use the same denominator for the mixed fraction.

convertIF:

First, we would receive an iFraction object as an input. Then we use a set of if conditions in order to convert this object into a Fraction object.

If the decimal part is zero, we simply have the numerator and the denominator of the mixed fraction be equal to the numerator and the denominator of the improper fraction respectively. If the decimal part is not zero, the numerator of the improper fraction would be equal to the multiplication of the decimal part and denominator added to the numerator while the denominator stays the same for both improper and mixed fractions.

showFraction function:

This function is used to display iFraction objects. Here we get an iFraction object as an input and separate its decimal part, numerator and denominator. Now, we would print the following to display the mixed fraction: decimal part(numerator/denominator).

The explanation of the algorithm of each of the nine options, which are included in the menu, is provided below:

- 1) Find the sum
- 2) Find the difference
- 3) Find the multiplication
- 4) Find the division
- 5) Convert Fraction to decimal
- 6) Convert decimal to Fraction
- 7) Compare two Fractions

For the seven options mentioned above, first we convert the input iFraction object into a Fraction object using the convertIF function. Then we use the seven equivalent functions (functions with the same name) that were used for Fraction objects in order to find the desired result. Next, we would use the convertF function to convert the result, which is a Fraction object, to an iFraction object and use the showFraction function to display the results on the screen.

It should be mentioned that, every time that the user chooses one of the mentioned options a message would prompt that asks them to choose between entering an improper or a mixed function. From there, the program takes two paths depending on what the user chooses.

8) Convert improper fraction to mixed fraction

We simply use the convertF function. The algorithm of this function has been explained above.

9) Convert mixed fraction to improper fraction

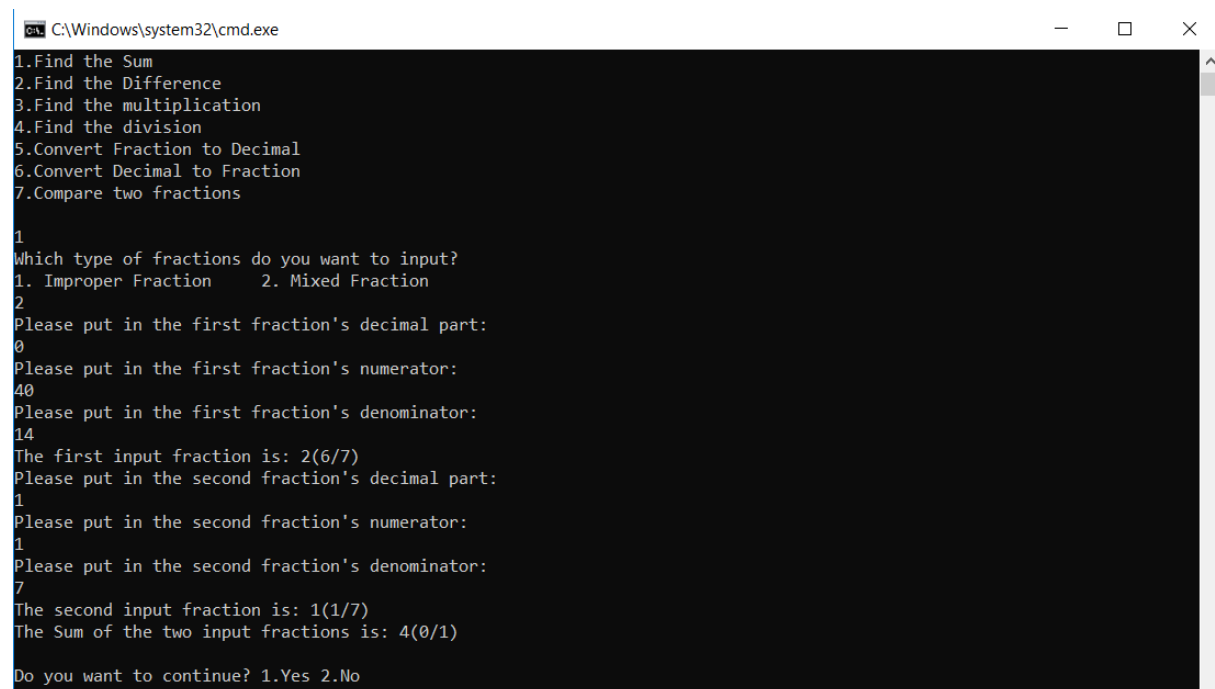
We simply use the convertIF function. The algorithm of this function has been explained above.

Implementation

Name of the file for this exercise ==> Assignment2_Exercise1. Cpp, Fraction.h and iFraction.h

Testing

For the testing of this program, every option available in the menu was tested with a number of different possible inputs in order to ensure the complete functionality of this program. The following nine figures (Figure 1-9) were instances of such tests. Different signs and different relations between the decimal part, the numerator and the denominator were used in order to ensure that all parts such as normalise function, convertF and convertIF functions work as well as the options provided in the menu.



```
C:\Windows\system32\cmd.exe
1.Find the Sum
2.Find the Difference
3.Find the multiplication
4.Find the division
5.Convert Fraction to Decimal
6.Convert Decimal to Fraction
7.Compare two fractions
1
Which type of fractions do you want to input?
1. Improper Fraction    2. Mixed Fraction
2
Please put in the first fraction's decimal part:
0
Please put in the first fraction's numerator:
40
Please put in the first fraction's denominator:
14
The first input fraction is: 2(6/7)
Please put in the second fraction's decimal part:
1
Please put in the second fraction's numerator:
1
Please put in the second fraction's denominator:
7
The second input fraction is: 1(1/7)
The Sum of the two input fractions is: 4(0/1)
Do you want to continue? 1.Yes 2.No
```

Figure 1: testing the plus function, showFraction, convertF and convertIF

```
C:\Windows\system32\cmd.exe
5.Convert Fraction to Decimal
6.Convert Decimal to Fraction
7.Compare two fractions
8.Convert improper fraction to mixed fraction
9.Convert mixed fraction to improper fraction
2
Which type of fractions do you want to input?
1. Improper Fraction    2. Mixed Fraction
2
Please put in the first fraction's decimal part:
1
Please put in the first fraction's numerator:
12
Please put in the first fraction's denominator:
7
The first input fraction is: 2(5/7)
Please put in the second fraction's decimal part:
2
Please put in the second fraction's numerator:
0
Please put in the second fraction's denominator:
0
Invalid denominator!
The denominator has been automatically set to 1 as a result!
The second input fraction is: 2(0/1)
The subtraction of the two input fractions is: 0(5/7)
Do you want to continue? 1.Yes 2.No
```

Figure 2: testing the minus function, showFraction, convertF and convertIF

```
C:\Windows\system32\cmd.exe
3.Find the multiplication
4.Find the division
5.Convert Fraction to Decimal
6.Convert Decimal to Fraction
7.Compare two fractions
8.Convert improper fraction to mixed fraction
9.Convert mixed fraction to improper fraction
3
Which type of fractions do you want to input?
1. Improper Fraction    2. Mixed Fraction
2
Please put in the first fraction's decimal part:
3
Please put in the first fraction's numerator:
0
Please put in the first fraction's denominator:
5
The first input fraction is: 3(0/1)
Please put in the second fraction's decimal part:
4
Please put in the second fraction's numerator:
0
Please put in the second fraction's denominator:
4
The second input fraction is: 4(0/1)
The Multiplication of the two input fractions is: 12(0/1)
Do you want to continue? 1.Yes 2.No
```

Figure 3: testing the times function, showFraction, convertF and convertIF

```
C:\Windows\system32\cmd.exe
3.Find the multiplication
4.Find the division
5.Convert Fraction to Decimal
6.Convert Decimal to Fraction
7.Compare two fractions
8.Convert improper fraction to mixed fraction
9.Convert mixed fraction to improper fraction
4
Which type of fractions do you want to input?
1. Improper Fraction    2. Mixed Fraction
2
Please put in the first fraction's decimal part:
-7
Please put in the first fraction's numerator:
8
Please put in the first fraction's denominator:
9
The first input fraction is: -6(-1/9)
Please put in the second fraction's decimal part:
6
Please put in the second fraction's numerator:
5
Please put in the second fraction's denominator:
4
The second input fraction is: 7(1/4)
The division of the two input fractions is: 0(-220/261)
Do you want to continue? 1.Yes 2.No
```

Figure 4: testing the division function, showFraction, convertF and convertIF

```
C:\Windows\system32\cmd.exe
Please Choose one of the following choices by entering its number :)
1.Find the Sum
2.Find the Difference
3.Find the multiplication
4.Find the division
5.Convert Fraction to Decimal
6.Convert Decimal to Fraction
7.Compare two fractions
8.Convert improper fraction to mixed fraction
9.Convert mixed fraction to improper fraction
5
Which type of fractions do you want to input?
1. Improper Fraction    2. Mixed Fraction
2
Please put in the fraction decimal part:
-7
Please put in the fraction numerator:
6
Please put in the fraction denominator:
9
The input fraction is: -6(-1/3)
The decimal form of this fraction is: -6.33333
Do you want to continue? 1.Yes 2.No
```

Figure 5: testing the convertToDecimal function, showFraction, convertF and convertIF

```
C:\Windows\system32\cmd.exe

Please Choose one of the following choices by entering its number :)
1.Find the Sum
2.Find the Difference
3.Find the multiplication
4.Find the division
5.Convert Fraction to Decimal
6.Convert Decimal to Fraction
7.Compare two fractions
8.Convert improper fraction to mixed fraction
9.Convert mixed fraction to improper fraction
5
Which type of fractions do you want to input?
1. Improper Fraction    2. Mixed Fraction
2
Please put in the fraction decimal part:
5
Please put in the fraction numerator:
4
Please put in the fraction denominator:
8
The input fraction is: 5(1/2)
The decimal form of this fraction is: 5.5

Do you want to continue? 1.Yes 2.No
```

Figure 6: testing the convertToFraction function, showFraction, convertF and convertIF

```
C:\Windows\system32\cmd.exe

3.Find the multiplication
4.Find the division
5.Convert Fraction to Decimal
6.Convert Decimal to Fraction
7.Compare two fractions
8.Convert improper fraction to mixed fraction
9.Convert mixed fraction to improper fraction
7
Which type of fractions do you want to input?
1. Improper Fraction    2. Mixed Fraction
2
Please put in the first fraction's decimal part:
7
Please put in the first fraction's numerator:
6
Please put in the first fraction's denominator:
5
The first input fraction is: 8(1/5)
Please put in the second fraction's decimal part:
9
Please put in the second fraction's numerator:
2
Please put in the second fraction's denominator:
7
The second input fraction is: 9(2/7)
First fraction is smaller than the second fraction!
In other words  $8(1/5) < 9(2/7)$ 
Do you want to continue? 1.Yes 2.No
```

Figure 7: testing the comparison of two functions, showFraction, convertF and convertIF


```
C:\Windows\system32\cmd.exe
Please Choose one of the following choices by entering its number :)
1.Find the Sum
2.Find the Difference
3.Find the multiplication
4.Find the division
5.Convert Fraction to Decimal
6.Convert Decimal to Fraction
7.Compare two fractions
8.Convert improper fraction to mixed fraction
9.Convert mixed fraction to improper fraction

8
Please put in the fraction numerator:
-4
Please put in the fraction denominator:
3
The input improper fraction is: -4/3
The output mixed fraction is: -1(-1/3)
Do you want to continue? 1.Yes 2.No
```

Figure 8: testing the showFraction and convertF functions

```
C:\Windows\system32\cmd.exe
Glad to hear that!
Please Choose one of the following choices by entering its number :)
1.Find the Sum
2.Find the Difference
3.Find the multiplication
4.Find the division
5.Convert Fraction to Decimal
6.Convert Decimal to Fraction
7.Compare two fractions
8.Convert improper fraction to mixed fraction
9.Convert mixed fraction to improper fraction

9
Please put in the fraction decimal part:
-6
Please put in the fraction numerator:
5
Please put in the fraction denominator:
7
The input mixed fraction is: -5(-2/7)
The output improper fraction is: -37/7
Do you want to continue? 1.Yes 2.No
```

Figure 9: testing the showFraction and convertIF functions