Лабораторная работа №3:
представление вещественных чисел

Цель:

Целью данной работы является получение базовых навыков работы с двоичным представлением вещественных чисел, на примере типа данных float

Справочная информация:

В языке C#, получить **побитовое представление вещественного числа** можно при помощи метода `BitConverter.GetBytes()`:

```
float n = 10.05;
byte[] tmp = BitConverter.GetBytes(n);
```

После завершения работы метода, в массиве tmp, окажется 4 байта (32 бита), из которых состоит число, переданное в качестве параметра. При этом, байты числа будут записаны **от младшего к старшему**, то есть tmp[0] будет содержать младший байт числа (последние 8 бит), а tmp[3] будет содержать старший байт числа (первые 8 бит).

Соответственно, переведя эти 4 байта в двоичную форму по правилам перевода в двоичную форму целых чисел, можно получить двоичное представление вещественного числа одинарной точности.

Вещественные числа в компьютерах различных типов записываются по-разному, тем не менее, все компьютеры поддерживают несколько международных стандартных форматов, различающихся по точности, но имеющих одинаковую **структуру следующего вида** (для одинарной точности - float):

Пример:

Исходное число: N = 10,05;

Результат `GetBytes()`: [205][204][32][65]

Результат конвертирования полученных байт к побитовому представлению:

205 -> 1100 1101

204 -> 1100 1100

32 -> 0010 0000

65 -> 0100 0001

Запись бит от старшего к младшему: 01000001 00100000 11001100 11001101

Знак (0й бит): 0

Порядок (1-8 биты): 10000010

Мантисса (9+ биты) 01000001100110011001101

Получить **вещественное число в десятичном виде**, имея его битовое представление, можно при помощи метода `BitConverter.ToSingle()`:

```
float f = BitConverter.ToSingle(res, 0);
```

где:

res – массив, содержащий 4 байта числа, от младшего к старшему.

0 – индекс массива, начиная с которого будет происходить преобразование.

Получить вещественное число в виде 4х байтов можно разбив его побитовое представление на блоки по 8 бит и преобразовав каждый из них в число типа byte, используя алгоритм преобразования целых чисел из двоичного вида в десятичный.

Пример:

Исходная строка: N = “01000001001000001100110011001101”;

Строка, разбитая на блоки по 8 бит: “01000001” “00100000” “11001100” “11001101”

Результат конвертирования блоков в целые числа:

“01000001” -> 65

“00100000” -> 32

“11001100” -> 204

“11001101” -> 205

Запись бит от старшего к младшему: [205][204][32][65]

Результат работы метода `ToSingle()`: 10,05

Алгоритмический перевод целой части десятичной дроби осуществляется таким же образом, как и перевод целого числа. Перевод дробной части осуществляется путём её домножения на 2 и взятия целой части результата.

Пример: Требуется перевести дробное десятичное число 206,116 в дробное двоичное число.

Перевод целой части дает $206_{10}=1100111_2$ по ранее описанным алгоритмам. Дробную часть двоичного числа:

```
.116 • 2 = 0.232
.232 • 2 = 0.464
.464 • 2 = 0.928
.928 • 2 = 1.856
.856 • 2 = 1.712
.712 • 2 = 1.424
.424 • 2 = 0.848
.848 • 2 = 1.696
.696 • 2 = 1.392
.784 • 2 = 0.784
```

Результат: $206,116_{10}=11001110,0001110110_2$

Вещественные числа принято хранить в виде: $x = M*q^p$

Где: M – значимая часть, q – основание системы исчисления, а p – степень.

Кроме того, числа должны быть представлены в нормализованном виде: $1 <= |M| < q$

Где: |M| - модуль мантиссы, а q – основание системы исчисления.

Соответственно, результат будет представлен как: $206{,}116_{10} = 1{,}10011100001110110_2 * 2^7$

В данном случае, порядок смещается на 127, и первые 127 значений служат для хранения отрицательных чисел.

Таким образом, в памяти компьютера, результат будет записан как: 0 00000111 10011100001110110

Перевод двоичного вещественного числа в десятичное осуществляется практически так же, как и для целых чисел:

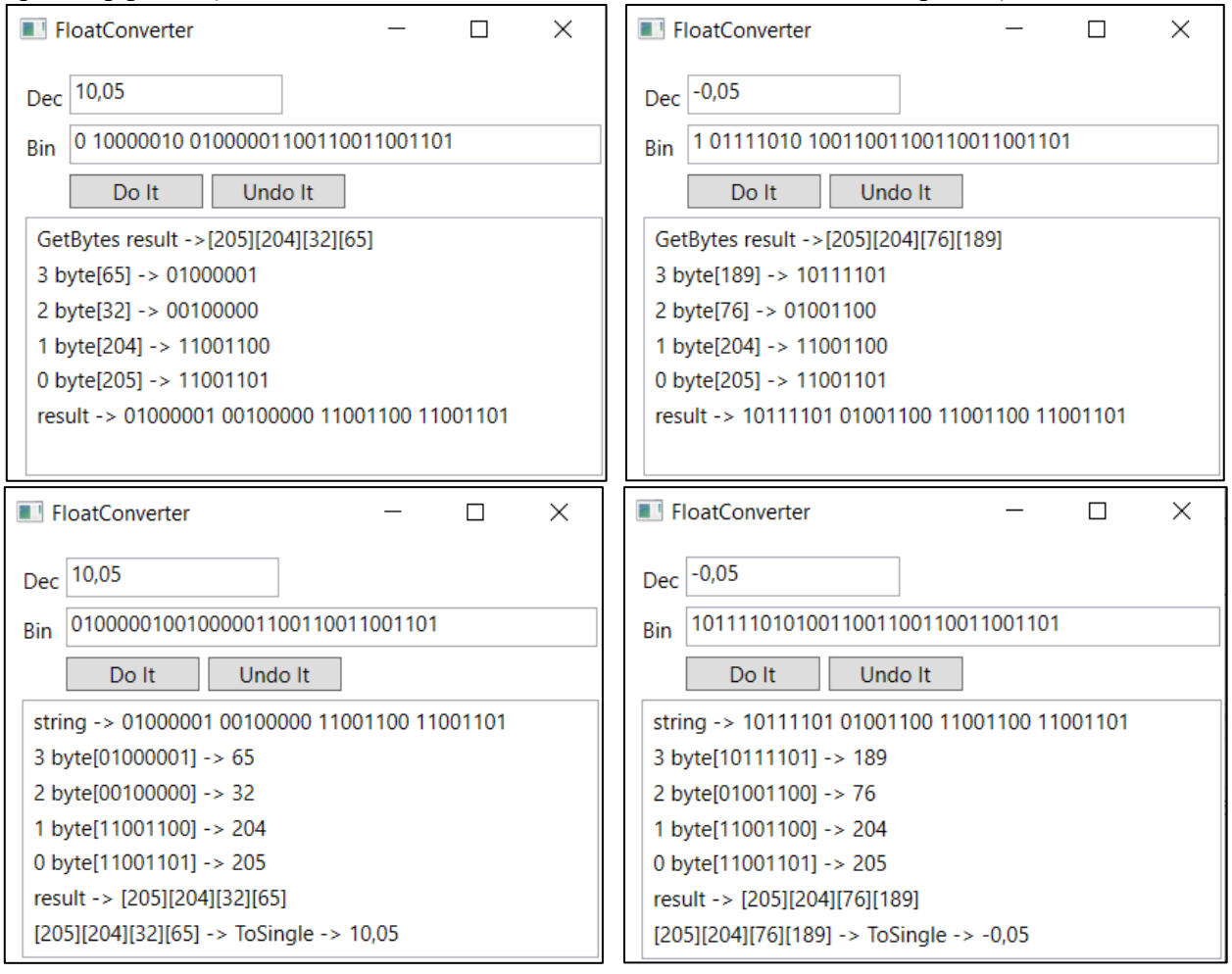
```
101,10 -> 1*2^2 + 0*2^1 + 1*2^0 + 1*2^-1 + 0*2^-2 = 16 + 0 + 4 + 2 + 0 = 5,5
101,10_2 = 5,5_10
```

Задание:

Разработайте и реализуйте программное приложение WPF, способное:

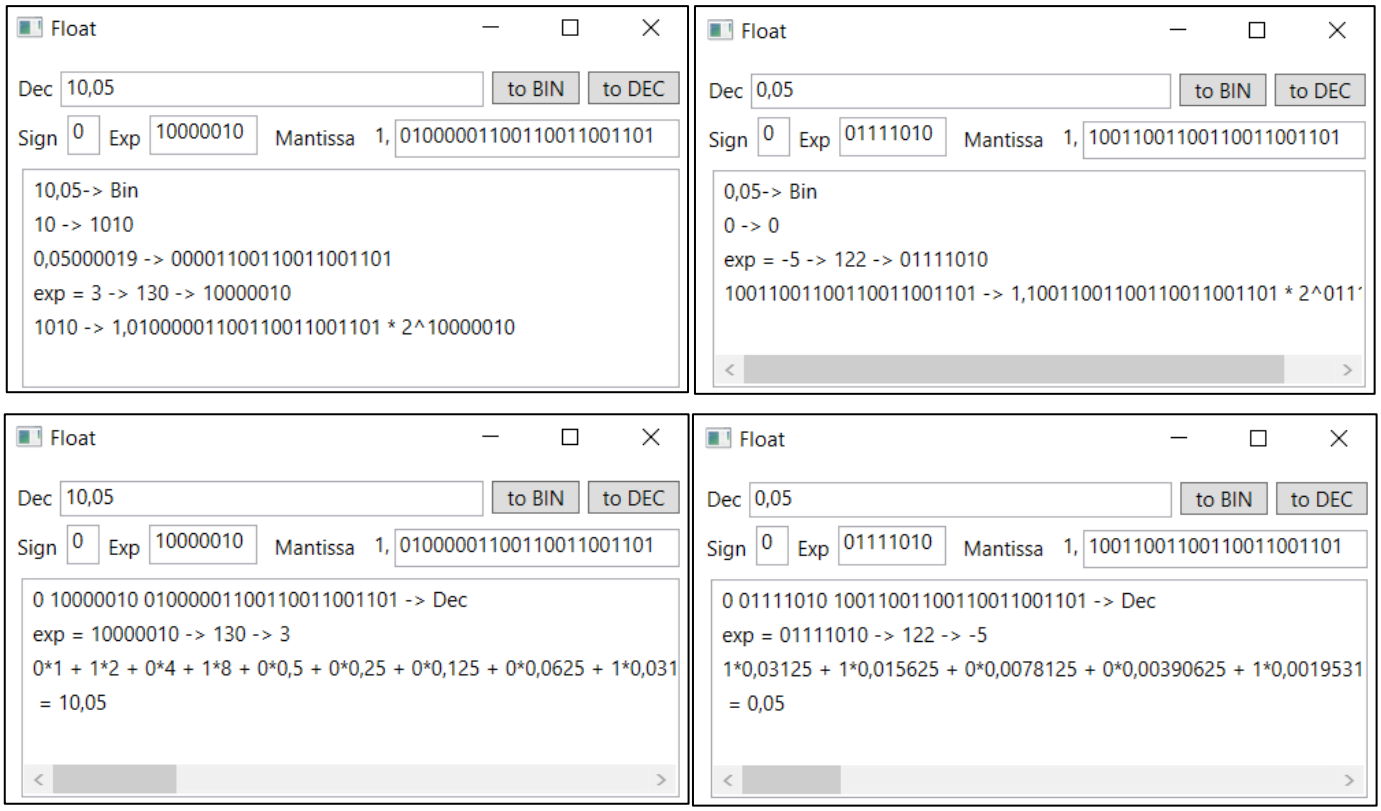
1. Осуществлять конвертирование десятичного представления вещественного числа в двоичное при помощи метода `BitConverter.GetBytes()`.
2. Осуществлять конвертирование двоичного представления вещественного числа в десятичное при помощи метода `BitConverter.ToSingle();`

Пример интерфейса (в нижней области выведена последовательность операций):



3. Осуществлять алгоритмическое конвертирование десятичного представления вещественного числа в двоичное.
4. Осуществлять алгоритмическое конвертирование двоичного представления вещественного числа в десятичное.

Пример интерфейса (в нижней области выведена последовательность операций):



Примечание: для представления чисел можно использовать массив вида:

```
byte[] a = new byte[32];
```

где:
элемент 0 – хранит знак
элементы 1-8 – хранят порядок
элементы 9-32 – хранят мантиссу

Список литературы:

Перевод вещественных чисел в двоичную систему онлайн: https://math.semestr.ru/inf/ieee754.php
Метод GetBytes:
https://docs.microsoft.com/en-us/dotnet/api/system.bitconverter.getbytes?view=net-5.0#System_BitConverter_GetBytes_System_Single_

Метод ToSingle:

https://docs.microsoft.com/en-us/dotnet/api/system.bitconverter.tosingle?view=net-5.0#System_BitConverter_ToSingle_System_Byte___System_Int32_

Двоичные числа и двоичная арифметика https://www.intuit.ru/studies/courses/685/541/lecture/12186

Арифметические основы компьютеров: http://book.kbsu.ru/theory/chapter4/1_4.html