1. Objective
The objective of this project is to learn the basic representations of floating points before simulating the pipelined processor. To simplify, we assume the following:
(1) Mantissa is 8bit
(2) Exponent is 8bit and uses excess-127 code (i.e., E + 127)
(3) All floating point numbers are positive.
It will be called 8-bit floating-point (FP) representation.

2. Conversion Program
Writing of the conversion program starts with creating below GUI using the C# ToolBox. The numbers in the textboxes should be empty, and the left side textboxes are for user entry input boxes.

For simplicity we do not use the sign bit and both mantissa and exponents are 8 bits, i.e., it can be structured.

```csharp
struct myfloat
{
    public byte mantissa;
    public byte exponent;
}
```

User enters a decimal number such as 25.7 in the Decimal textbox, and clicking the “To Float Point” button should fill in the Mantissa and Exponent textboxes in a binary string form. Also, when mantissa and exponent are supplied in binary strings, clicking the “To Decimal” button should produce an equivalent decimal number in the right side textbox.
Suppose that you named the Decimal textbox as txtDecimal, then read-in of the textbox can be done using:

```csharp
float inData;
if (float.TryParse(txtDecimal.Text, out inData))
{
    //inData contains the converted value
}
else
{
    MessageBox.Show("Enter a number in the Decimal textbox!");
    return;
}
```

You should always use “type.TryParse()” when you read in user inputs because of potential user entry errors. Next, you will need to separate the integer and fraction portion of the number inData and perform a proper conversion process.

For displaying binary strings in textboxes, you may use the `Convert.ToString()` function, i.e.,

```csharp
byte b1=56;
txt1.Text += Convert.ToString(b1, 2);
```

The second parameter “2” indicates the base 2.

After completing the program, you may check the correctness by converting from Decimal to FP and then vice versa. Due to only 8-bits used in mantissa, rounding errors will be visible.

3. Check-off

Program check-off is required. You may bring your program in a flash memory stick.

You may check if your conversion runs correctly using the following web site:

4. Report

1. A short intro on what you are doing in this project.
2. Show an example of manual step-by-step conversion from a decimal number to 8-bit floating point representation. For example, 31.12.
3. Show an example of manual step-by-step conversion from an 8-bit floating point number to decimal representation. Convert back from answer of 2.
4. Code list