(1) What does the following 16-bit two’s complement number mean in the decimal format? Show your work (how you calculated your answer).

```
1 0 0 1 0 0 0 0 0 1 1 0 1 0 1
```

\[-2^{15} + 2^{12} + 2^5 + 2^4 + 2^2 + 2^0\]

\[-28,619\]

(2) What is “Overflow Flag” in a processor (how is it used) for handling operations on two’s complement integers?

The overflow flag in a processor is raised when the results of a (two’s complement) math operation (calculation) either exceeds the largest positive number the processors can get (using two’s complement integer) or goes lower than the smallest negative number the processor can get (using the two’s complement integer).

Note: “The overflow flag in a processor is raised when the results of a (two’s complement) math operation (calculation) generates an extra bit to fit (the number of the bits the processor has)” is not a correct solution for this question (that’s what the carry flag is for).
(3) Assuming 4-bit two’s complement integers, will the following calculation cause:

(a) The overflow flag. If yes, show how. If no, show how not.
(b) The carry flag. If yes, show how. If no, show how not.

\[
\begin{array}{c}
1 & 1 & 1 & 0 \\
+ & 0 & 0 & 1 & 0 \\
\end{array}
\]

**Solutions:**

- Using the four-bit two’s complement integer, the largest positive a processor can get: +7
- Using the four-bit two’s complement integer, the smallest negative a processor can get: -8

(a) **NO:** Since adding “1110” (which means ‘-2’) to “0010” (which means ‘2’) will not produce an incorrect result (i.e., the processor did not go across “the 6:00 o’clock and 5:00 o’clock boundary), it will not raise the overflow flag.

(b) **YES:** Since the calculation will generate one additional bit (as shown below), the carry flag will be raised.

\[
\begin{array}{c}
1 & 1 & 1 & 0 \\
+ & 0 & 0 & 1 & 0 \\
\hline
1 & 0 & 0 & 0 & 0 \\
\end{array}
\]

(4) What are the three components in the floating-point number expressions (i.e., name the three “components”)?

- The sign bit
- The exponent bits
- The significand (mantissa) bits
(5) Translate $+181.625_{(10)}$ in the normalized binary format. Show your work.

**Note 1:** This question does not ask you to transform the number in “the bias of 127” or “the complete IEEE-754 floating-point fraction number”.

**Note 2:** You do not need to translate “exponent” using “bias of 127” (you can specify the exponent of 2 using a decimal number)

\[
181 = 1011010101 \\
.625 = (1/2) + (1/8) = .101
\]

\[
181.625 = 10110101.101 = 10110101.101 \times 2^0 \\
= 1.0110101101 \times 2^7
\]