1. Convert $4096_{10}$, $-2047_{10}$, and $-2000000_{10}$ into 32-bit two’s complement binary numbers, respectively, and convert the following two’s complement binary numbers to be decimal numbers:

   a. 1111 1111 1111 1111 1111 1111 0000 0110\text{two};
   b. 1111 1111 1111 1111 1111 1111 1110 1111\text{two};
   c. 0111 1111 1111 1111 1111 1111 1110 1111\text{two}.

2. Suppose that all of the conditional branch instructions except \textit{beq} and \textit{bne} were removed from the MIPS instruction set along with \textit{slt} and all of its variants (\textit{slti}, \textit{sltu}, \textit{sltui}). Show how to perform

   \begin{verbatim}
   slt $t0, $s0, $s1
   \end{verbatim}

   using the modified instruction set in which \textit{slt} is not available. (Hint: It requires more than two instructions.)

3. The ALU supported set on less than (\textit{slt}) using just the sign bit of the adder. Let’s try a set on less than operation using the values $-7_{10}$ and $6_{10}$. To make it simpler to follow the example, let’s limit the binary representations to 4 bits: $1001_{two}$ and $0110_{two}$.

   $1001_{two} - 0110_{two} = 1001_{two} + 1010_{two} = 0011_{two}$

   This result would suggest that $-7_{ten} > 6_{ten}$, which is clearly wrong. Hence we must factor in overflow in the decision. Modify the 1-bit ALU in the following figures to handle \textit{slt} correctly.

![Figure 1: A 1-bit ALU that performs AND, OR, and addition on a and b or b'.](image-url)
Figure 2: A 1-bit ALU for the most significant bit.

4. Add $2.85_{ten} \times 10^3$ to $9.84_{ten} \times 10^4$ and add $3.63_{ten} \times 10^4$ to $6.87_{ten} \times 10^3$, respectively, assuming that you have only three significant digits, first with guard and round digits and then without them.

5. Show the IEEE 754 binary representation for the floating-point numbers $20_{ten}$, $20.5_{ten}$, $0.1_{ten}$, and $-5/6$, respectively.