1. Convert $512_{\text{ten}}$, $-1,023_{\text{ten}}$, and $-4,000,000_{\text{ten}}$ 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 1110 0000 1100$_{\text{two}}$;
   b. 1111 1111 1111 1111 1111 1111 1111 1111$_{\text{two}}$;
   c. 0111 1111 1111 1111 1111 1111 1111 1111$_{\text{two}}$.

2. Find the shortest sequence of MIPS instructions to determine the absolute value of a two’s complement integer. Convert this instruction (accepted by the MIPS assembler):

   ```
   abs $t2, $t3
   ```

   This instruction means that register $t2$ has a copy of register $t3$ if register $t3$ is positive, and the two’s complement of register $t3$ if $t3$ is negative. (Hint: It can be done with three instructions.)

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

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

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

![Figure 1: A 1-bit ALU that performs AND, OR, and addition on a and b or b’](image-url)
4. Add $2.85_{10} \times 10^3$ to $9.84_{10} \times 10^4$ and add $3.63_{10} \times 10^4$ to $6.87_{10} \times 10^3$, 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 number $10_{10}$, $10.5_{10}$, $0.1_{10}$, and $-2/3$, respectively.