

Problem C5. (Unified Computers and programming)

1. Convert the following base 10 numbers into 8-bit 2's complement notation

0, -1, -12

2. Perform each of the following additions assuming that the bit strings represent values in 2's complement notation. Identify the cases in which the answer is incorrect because of overflow.

$$\begin{array}{r} 1111 \\ + 1111 \\ \hline \end{array} \quad \begin{array}{r} 01111 \\ + 10001 \\ \hline \end{array} \quad \begin{array}{r} 01110 \\ + 01010 \\ \hline \end{array}$$

Hint: The bit strings are only 5 bits long so your answer should also be 5 bits long.

3. Write an algorithm to convert a negative decimal number into a binary number in 2's complement form. Assume that the number ranges from +127 to -128

Hint: You already know how to convert a positive decimal number into binary notation. Think about determining sign and inverting bit positions.

4. Implement your algorithm in Ada95. Turn in an electronic copy of your code listing and a hard copy of your code.