Problem 1. Encoding positive integers

1. What is the 5-bit binary representation of the decimal number 21? ★

2. What is the hexadecimal representation for decimal 219 encoded as an 8-bit binary number?

3. What is the hexadecimal representation for decimal 51 encoded as a 6-bit binary number?

4. The hexadecimal representation for an 8-bit unsigned binary number is 0x9E. What is its decimal representation? ★

5. What is the range of integers that can be represented with a single unsigned 8-bit quantity?

6. Since the start of official pitching statistics in 1988, the highest number of pitches in a single game has been 172. Assuming that remains the upper bound on pitch count, how many bits would we need to record the pitch count for each game as an unsigned binary number?

7. Compute the sum of these two 4-bit unsigned binary numbers. Express the result in hexadecimal. ★

\[
\begin{align*}
1101 \\
+0110
\end{align*}
\]
Problem 2. Two’s complement representation

1. What is the 6-bit two’s complement representation of the decimal number -21?

2. What is the hexadecimal representation for decimal -51 encoded as an 8-bit two’s complement number?

3. The hexadecimal representation for an 8-bit two’s complement number is 0xD6. What is its decimal representation?

4. Using a 5-bit two’s complement representation, what is the range of integers that can be represented with a single 5-bit quantity?

5. Can the value of the sum of two 2’s complement numbers 0xB3 + 0x47 be represented using an 8-bit 2’s complement representation? If so, what is the sum in hex? If not, write NO.

6. Can the value of the sum of two 2’s complement numbers 0xB3 + 0xB1 be represented using an 8-bit 2’s complement representation? If so, what is the sum in hex? If not, write NO.
7. Please compute the value of the expression 0xBB – 8 using 8-bit two’s complement arithmetic and give the result in decimal (base 10).

8. Consider the following subtraction problem where the operands are 5-bit two’s complement numbers. Compute the result and give the answer as a decimal (base 10) number. ★

\[
\begin{array}{c}
10101 \\
- 00011
\end{array}
\]
Problem 3. Multiples of 4

1. Given an unsigned $n$-bit binary integer $v = b_{n-1} \ldots b_1 b_0$, prove that $v$ is a multiple of 4 if and only if $b_0 = 0$ and $b_1 = 0$.

2. Does the same relation hold for two’s complement encoding?
**Problem 4. Encoding text**

There are multiple standards to encode characters and strings using binary values. ASCII is a classic standard to encode English alphabet characters (modern formats like UTF support other alphabets, but are typically based on ASCII). ASCII encodes each character using an 8-bit (1-byte) value. The table below shows ASCII’s mapping of characters to values.

| A   | B   | C   | D   | E   | F   | G   | H   | I   | J   | K   | L   | M   | N   | O   |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0   | NUL | SOH | STX | ETX | EOT | ENQ | ACK | BEL | BS  | HT  | LF  | VT  | FF  | CR  | SO  | SI  |
| 1   | DLE | DC1 | DC2 | DC3 | DC4 | NAK | SYN | ETB | CAN | EM  | SUB | ESC | FS  | GS  | RS  | US  |
| 2   | !   | "   | #   | $   | %   | &   | ’   | (   | )   | *   | +   | ,   | -   | .   | /   |
| 3   | \n | 0x20 | 0x21 | 0x22 | 0x23 | 0x24 | 0x25 | 0x26 | 0x27 | 0x28 | 0x29 | 0x2A | 0x2B | 0x2C | 0x2D | 0x2E | 0x2F |
| 4   | @   | A   | B   | C   | D   | E   | F   | G   | H   | I   | J   | K   | L   | M   | N   | O   |
| 5   | P   | Q   | R   | S   | T   | U   | V   | W   | X   | Y   | Z   | [   | \\ | ]   | ^   | _   | `   |
| 6   | \n | 0x30 | 0x31 | 0x32 | 0x33 | 0x34 | 0x35 | 0x36 | 0x37 | 0x38 | 0x39 | 0x3A | 0x3B | 0x3C | 0x3D | 0x3E | 0x3F |
| 7   | p   | q   | r   | s   | t   | u   | v   | w   | x   | y   | z   | {   | |   | |   | ~   | DEL |

| Letter | Number | Punctuation | Symbol | Other/non-printable |

Computers often store variable-length text as a null-terminated string: a sequence of bytes, where each byte denotes a different character, terminated by the value 0x00 (null) to denote the end of the string. For example, the string “6.004” is encoded as the 6-byte sequence 0x362E30303400.

1. Encode your name as a null-terminated ASCII string (use the best approximation if your name contains non-English characters)

2. Decode the following null-terminated ASCII string:

0x 52 49 53 43 2D 56 20 69 73 20 63 6F 6D 69 6E 67 21 00