

Binary Numbers & Number Systems

Name: _____ Period: _____ Date: _____

Objectives

- Explain how computing devices represent all data using bits (DAT-1.A).
- Convert between binary (base 2) and decimal (base 10), and compare binary numbers (DAT-1.C).
- Explain how a fixed number of bits leads to overflow and round-off errors (DAT-1.B).

Directions: Work through each section. Show your place-value work on conversions — the exam rewards reasoning, not just the answer.

Section 1: Bits, Bytes, and Abstraction

Fill in the blank or answer briefly.

1. A _____ is the smallest unit of information in a computer and is either _____ or _____.
2. "Bit" is short for _____.
3. A byte is _____ bits.
4. The same sequence of bits (for example 01000001) can represent different types of data depending on context. Give two different things a byte might represent.

5. Define **abstraction** in one sentence, then explain how grouping bits to represent a number is an example of it.

Section 2: Binary Place Values

The place value of each position is the base (2) raised to the power of the position, numbered from the right starting at 0. Fill in the missing place values for an 8-bit number.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128							

Section 3: Binary → Decimal

Convert each binary number to decimal.

6. 1011 Decimal: _____

Show your work: _____

7. 101101 Decimal: _____

Show your work: _____

8. 11110000 Decimal: _____

Show your work: _____

Section 4: Decimal → Binary

Convert each decimal number to binary.

9. 19 Binary: _____

Show your work: _____

10. 42 Binary: _____

Show your work: _____

11. 200 Binary: _____

Show your work: _____

Section 5: Comparing & Ordering Binary Numbers

Answer each question.

12. Which is larger, 1001 or 1010? Convert both to justify your answer.

13. Order these binary numbers from least to greatest: 1100, 1001, 1010, 0111.

Section 6: Overflow & Round-off Errors

Apply what you know about representing values with a fixed number of bits.

14. An 8-bit unsigned integer can store values from 0 to ?. What is the largest value an 8-bit unsigned integer can store, and what happens if a program tries to compute that value +1 in 8 bits? Name this type of error.

15. Explain why some real numbers (like $1/3$) cannot be stored exactly in a fixed number of bits. Name this type of error.
