بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Quran Education Point

QuranEducationPoint.Com

Class	9th	Chapter	02
Subject	Computer Science	Board	Punjab

CHAPTER 2: NUMBER SYSTEMS

Exercise Solution

PART I: MULTIPLE-CHOICE QUESTIONS (MCQs)

1. What does ASCII stand for?

a) American Standard Code for Information Interchange | b) Advanced Standard Code for Information Interchange | c) American Standard Communication for Information Interchange | d) Advanced Standard Communication for Information Interchange

Answer: a) American Standard Code for Information Interchange

2. Which of the following numbers is a valid binary number?

a) (110110_2) | b) (11011_2) | c) (110.11_2) | d) (1101A_2)

Answer: b) (11011_2)

3. How many bits are used in the standard ASCII encoding?

a) 7 bits | b) 8 bits | c) 16 bits | d) 32 bits

Answer: b) 8 bits (Note: Extended ASCII uses 8 bits; standard ASCII is 7-bit.)

4. Which of the following is a key advantage of Unicode over ASCII?

a) It uses fewer bits per character | b) It can represent characters from many different languages | c) It is backward compatible with binary | d) It is specific to the English language

Answer: b) It can represent characters from many different languages

5. How many bytes are used to store a typical integer?

a) 1 byte | b) 2 bytes | c) 4 bytes | d) 8 bytes

Answer: c) 4 bytes

6. What is the primary difference between signed and unsigned integers?

a) Unsigned integers cannot be negative | b) Signed integers have a larger range | c) Unsigned integers are stored in floating-point format | d) Signed integers are only used for positive numbers

Answer: a) Unsigned integers cannot be negative

7. In single precision, how many bits are used for the exponent?

a) 23 bits | b) 8 bits | c) 11 bits | d) 52 bits

Answer: b) 8 bits

8. What is the approximate range of values for single-precision floating-point numbers?

a) (1.4 \times 10^{-45}) to (3.4 \times 10^{38}) | b) (4.9 \times 10^{-324}) to (1.8 \times 10^{308})

Answer: a) (1.4 \times 10^{-45}) to (3.4 \times 10^{38})

QuranEducationPoint.Com

9. What are the tiny dots that make up an image called?
a) Pixels | b) Bits | c) Bytes | d) Nodes
Answer: a) Pixels
10. In an RGB color model, what does RGB stand for?
a) Red, Green, Blue | b) Red, Gray, Black | c) Right, Green, Blue | d) Red, Green, Brown
Answer: a) Red, Green, Blue

PART II: SHORT QUESTIONS
Q1. What is the primary purpose of the ASCII encoding scheme?
Answer:
ASCII (American Standard Code for Information Interchange) provides a standardized method to represent text characters, including letters, numbers, punctuation marks, and control codes, as numeric binary codes. This allows different computer hardware and software systems to exchange text data accurately.
Q2. Explain the difference between ASCII and Unicode.
Answer:
ASCII: ASCII is a 7-bit character encoding scheme that represents 128 characters. Extended ASCII uses 8 bits and can represent up to 256 values. ASCII primarily supports the English alphabet and basic symbols.
Unicode: Unicode is a universal character encoding standard that supports over a million possible code points, allowing it to represent text from virtually all global languages, scripts, and mathematical or emoji symbols.
Q3. How does Unicode handle characters from different languages?
Answer:
Unicode assigns a unique numerical code point, ranging from U+0000 to U+10FFFF, to every character across different writing systems. Encodings such as UTF-8, UTF-16, and UTF-32 translate these code points into binary byte sequences for computer processing and storage.
4. What is the range of values for an unsigned 2-byte integer?
Answer: A 2-byte integer contains 16 bits ($2^{16} = 65{,}536$ possible values). For an unsigned integer, the range is from 0 to 65,535.
5. Explain how a negative integer is represented in binary.
Answer: Negative integers are commonly represented using the Two's Complement method. To represent a negative number, take the positive binary value, invert all its bits (0s to 1s and 1s to 0s to get the one's complement), and add 1 to the least significant bit. The most significant bit (MSB) acts as the sign bit (1 indicates negative).
6. What is the benefit of using unsigned integers?
Answer: Unsigned integers do not reserve a bit for a sign flag; therefore, all available bits are dedicated entirely to magnitude. This doubles the maximum positive range achievable compared to signed integers of the same bit length.
7. How does the number of bits affect the range of integer values?

QuranEducationPoint.Com

Answer: The range of integer values grows exponentially with the number of bits (n). For unsigned integers, the range is 0 to $2^n - 1$. For signed integers, the range is -2^{n-1} to $2^{n-1} - 1$. Adding more bits expands the representable numerical span.

Q8. Why are whole numbers commonly used in computing for quantities that cannot be negative?

Answer:

Quantities such as memory addresses, pixel counts, array indices, and item quantities are generally represented as non-negative values. Using unsigned whole numbers for these metrics can optimize the available numerical range and help prevent logical errors.

9. How is the range of floating-point numbers calculated for single precision?

Answer: Single-precision floating-point numbers (following IEEE 754 standard) use 32 bits divided into 1 sign bit, 8 exponent bits, and 23 mantissa bits. The range is calculated based on the exponent limits (2^{-126} to 2^{127} approximately), yielding values ranging from roughly 1.4×10^{-45} to 3.4×10^{38}.

10. Why is it important to understand the limitations of floating-point representation in scientific computing?

Answer: Floating-point numbers have finite precision and cannot represent every real number exactly. Understanding these limitations prevents rounding errors, catastrophic cancellation, and precision loss in critical scientific, financial, and engineering calculations.

Long Questions

1. Explain how characters are encoded using Unicode. Provide examples of characters from different languages and their corresponding Unicode code points.

Answer:

- Encoding Mechanism: Unicode assigns a unique hexadecimal identifier called a code point (prefixed with U+) to every character. Variable-width encodings like UTF-8 use 1 to 4 bytes to represent these code points efficiently, maintaining backward compatibility with ASCII for English characters while using multiple bytes for other languages.
- Examples:
    - English letter 'A': Code point U+0041
    - Urdu letter 'الف' (Alif): Code point U+0627
    - Chinese character '中': Code point U+4E2D
    - Euro currency symbol '€': Code point U+20AC

2. Describe in detail how integers are stored in computer memory.

Answer:

- Integers are stored in fixed-size binary blocks (typically 1, 2, 4, or 8 bytes).
- Unsigned Integers: Stored in pure binary format representing absolute magnitudes.
- Signed Integers: Stored using Two's Complement representation. The leftmost bit serves as the sign bit (0 for positive, 1 for negative). Positive numbers are stored in standard binary, while negative numbers are stored as the two's complement of their absolute

QuranEducationPoint.com

value, allowing arithmetic units to perform subtraction using standard binary addition circuitry.

3. Explain the process of converting a decimal integer to its binary representation and vice versa. Include examples of both positive and negative integers.

Answer:

- Decimal to Binary (Positive): Successive division by 2, recording the remainders in reverse order (bottom-up).
  - *Example:* Convert 13_{10} to binary.

$13 \div 2 = 6$ remainder 1
$6 \div 2 = 3$ remainder 0
$3 \div 2 = 1$ remainder 1
$1 \div 2 = 0$ remainder 1
Reading remainders bottom-up: 1101_2.

- Binary to Decimal: Multiply each binary digit by 2 raised to the power of its position index (starting from 0 on the right) and sum the results.
  - *Example:* $1101_2 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 8 + 4 + 0 + 1 = 13_{10}$.
- Negative Integers: Convert the positive equivalent to binary, pad to the desired bit-width (e.g., 8 bits), invert all bits (1's complement), and add 1 (2's complement). For example, -13 in 8-bit two's complement is 11110011_2.

4. Perform the following binary arithmetic operations:

- a. Multiplication of 101_2 by 11_2:

$$\begin{array}{r} 101 \\ \times\ \ 11 \\ \hline 101 \\ 101 \\ \hline 1111_2 \end{array}$$

Result: 1111_2 (Decimal $5 \times 3 = 15$)

- b. Division of 1100_2 by 10_2:

$$1100_2 \div 10_2 = 110_2$$

Result: 110_2 (Decimal $12 \div 2 = 6$)

- c. Add the following binary numbers:
  - a) $101_2 + 110_2$:

$$\begin{array}{r} 101 \\ +110 \\ \hline 1011_2 \end{array}$$

Result: 1011_2 (Decimal $5 + 6 = 11$)

  - b) $1100_2 + 1011_2$:

QuranEducationPoint.Com

$$\begin{array}{r} 1100 \\ +1011 \\ \hline 10111_2 \end{array}$$

Result: 10111_2 (Decimal $12 + 11 = 23$)

5. Convert the following numbers to 4-bit binary and add them:

- (a) $7 + (-4)$:
  - $+7$ in 4-bit binary = 0111_2
  - -4 in 4-bit 2's complement = 1100_2
  - Addition: $0111 + 1100 = (1)0011_2$. Dropping the overflow carry bit yields 0011_2 (Decimal $+3$).
- (b) $-5 + 3$:
  - -5 in 4-bit 2's complement = 1011_2
  - $+3$ in 4-bit binary = 0011_2
  - Addition: $1011 + 0011 = 1110_2$.
  - In 2's complement, 1110_2 represents -2.

6. Solve the following binary subtractions:

- (a) $1101_2 - 0100_2$:

$$\begin{array}{r} 1101 \\ -0100 \\ \hline 1001_2 \end{array}$$

Result: 1001_2 (Decimal $13 - 4 = 9$)

- (b) $1010_2 - 0011_2$:

$$\begin{array}{r} 1010 \\ -0011 \\ \hline 0111_2 \end{array}$$

Result: 0111_2 (Decimal $10 - 3 = 7$)

- (c) $1000_2 - 0110_2$:

$$\begin{array}{r} 1000 \\ -0110 \\ \hline 0010_2 \end{array}$$

Result: 0010_2 (Decimal $8 - 6 = 2$)

- (d) $1110_2 - 100_2$:

$$\begin{array}{r} 1110 \\ -0100 \\ \hline 1010_2 \end{array}$$

Result: 1010_2 (Decimal $14 - 4 = 10$)

QuranEducationPoint.Com