Chapter 2: Number Systems

Short Question from the Chapter "Number System":

EXERCISE Short Question with Answers

Q1. What is the primary purpose of the ASCII encoding scheme?
Ans: ASCII is used to represent characters (letters, digits, and symbols) in computers.
It assigns a unique numeric code to each character for data storage and communication.

Q2. Explain the difference between ASCII and Unicode.
Ans: ASCII uses 7 or 8 bits and can represent only 128 or 256 characters.
Unicode uses more bits and supports thousands of characters from many languages.

Q3. How does Unicode handle characters from different languages?
Ans: Unicode assigns a unique code to each character, no matter the language.
It supports scripts like English, Arabic, Chinese, and many others in one system.

Q4. What is the range of values for an unsigned 2-byte integer?
Ans: A 2-byte integer has 16 bits.
For unsigned integers, the range is **0 to 65,535**.

Q5. Explain how a negative integer is represented in binary.
Ans: Negative integers are stored using **two's complement method**.
The most significant bit (MSB) is used as the sign bit: 0 for positive, 1 for negative.

Q6. What is the benefit of using unsigned integers?
Ans: Unsigned integers provide only positive values.
This increases the maximum range compared to signed integers.

Q7. How does the number of bits affect the range of integer values?
Ans: More bits increase the range of integers that can be stored.
For example, 8 bits store 0–255, while 16 bits store 0–65,535 (unsigned).

Q8. Why are whole numbers commonly used in computing for quantities that cannot be negative?
Ans: Whole numbers are simple and efficient to store.
They are used for quantities like age, students, or items, which can't be negative.

Q9. How is the range of floating-point numbers calculated for single precision?
Ans: Single precision uses 32 bits: 1 for sign, 8 for exponent, and 23 for mantissa.
The range depends on exponent bits, giving approx. **10^{-38} to 10^{38}**.

Q10. Why is it important to understand the limitations of floating-point representation in scientific computing?
Ans: Floating-point numbers cannot represent all real values exactly.
Understanding limitations helps avoid errors in precision-sensitive calculations.

Additional Short Question with Answer

Topic 1: Character Encoding (ASCII & Unicode)

Q1. Why was ASCII not enough for global communication?
Ans: ASCII supports only English characters.
It cannot represent characters of other languages like Arabic, Chinese, or Urdu.

Q2. How many characters can standard ASCII represent?
Ans: Standard ASCII uses 7 bits.
It can represent **128 characters** in total.

Q3. How many characters can extended ASCII represent?
Ans: Extended ASCII uses 8 bits.
It represents **256 characters**, including extra symbols.

Q4. Give one advantage of Unicode for international software.
Ans: Unicode allows consistent representation of all languages.
It ensures data can be shared across countries without errors.

Q5. What does a Unicode code point look like?
Ans: Unicode code points are written as **U+XXXX**.
For example, U+0041 represents the letter **A**.

Topic 2: Integer Representation

Q1. How many bytes are used for a 32-bit integer?
Ans: A 32-bit integer requires 4 bytes.
Since 1 byte = 8 bits, 4 × 8 = 32 bits.

Q2. What is the range of a signed 8-bit integer?
Ans: Signed 8-bit integers use two's complement.
The range is **–128 to +127**.

Q3. What is the range of an unsigned 8-bit integer?
Ans: Unsigned integers store only positive values.
The range is **0 to 255**.

Q4. Why do computers use two's complement for negative numbers?
Ans: It simplifies arithmetic operations.
The same hardware can add both positive and negative numbers.

Q5. What is the MSB (Most Significant Bit) used for in signed integers?
Ans: MSB is the **sign bit**.
0 means positive, and 1 means negative.

Topic 3: Floating-Point Representation

Q1. How many bits are used in single-precision floating-point numbers?
Ans: Single precision uses **32 bits**.
It includes 1 sign bit, 8 exponent bits, and 23 fraction bits.

Q2. How many bits are used in double-precision floating-point numbers?
Ans: Double precision uses **64 bits**.
It includes 1 sign bit, 11 exponent bits, and 52 fraction bits.

Q3. Why are floating-point numbers used in scientific calculations?
Ans: They represent very large or very small values.
This makes them suitable for scientific and engineering work.

Q4. What is the smallest non-zero positive number in single precision?
Ans: About $\mathbf{1.4 \times 10^{-45}}$.
This is the smallest positive value representable.

Q5. What is the largest number in single precision?
Ans: About $\mathbf{3.4 \times 10^{38}}$.
This is the maximum positive value stored.

Topic 4: Binary Arithmetic

Q1. What is the result of 101 + 11 in binary?
Ans: 101 (5) + 11 (3) = 1000.
So, the answer is **1000 (8 in decimal)**.

Q2. Perform 111 – 101 in binary.
Ans: 111 (7) – 101 (5) = 10.
So, the answer is **10 (2 in decimal)**.

Q3. Perform 10 × 11 in binary.
Ans: 10 (2) × 11 (3) = 110.
So, the answer is **110 (6 in decimal)**.

Q4. Perform 100 ÷ 10 in binary.
Ans: 100 (4) ÷ 10 (2) = 10.
So, the answer is **10 (2 in decimal)**.

Q5. Add 1010 and 111 in binary.
Ans: 1010 (10) + 111 (7) = 10001.
So, the answer is **10001 (17 in decimal)**.

Topic 5: Images & Colors (Pixels, RGB)

Q1. What is a pixel?
Ans: A pixel is the smallest dot of an image.
Thousands of pixels form a complete picture.

Q2. How many bits are used in true color representation?
Ans: True color uses **24 bits per pixel**.
8 bits for Red, 8 for Green, and 8 for Blue.

Q3. What does RGB stand for?
Ans: RGB stands for **Red, Green, Blue**.
These three colors mix to form all other colors.

Q4. What is grayscale image representation?
Ans: A grayscale image uses shades of black and white.
Each pixel stores brightness instead of full color.

Q5. Why are pixels important in digital graphics?
Ans: Pixels decide the **resolution** of an image.
More pixels = higher quality and sharper pictures.
