Class: 9th

Subject: Computer

Unit 2: Number Systems

❖ Important MCQs:

1. Which numbering system is used in everyday life?

(a) Binary

(b) Decimal ✅

(c) Octal

(d) Hexadecimal

2. The decimal number system is based on which base?

(a) Base-2

(b) Base-8

(c) Base-10 ✅

(d) Base-16

3. How many digits are used in the binary number system?

(a) 2 ✅

(b) 8

(c) 10

(d) 16

4. In binary system, the digit 1 represents:

(a) OFF

(b) ON ✅

(c) NULL

(d) ERROR

5. Which of the following is a valid binary number?

(a) 1021

(b) 1101 ✅

(c) 2101

(d) 1A01

6. What is the base of the hexadecimal number system?

(a) 2

(b) 8

(c) 10

(d) 16 ✅

7. Why do computers use the binary number system?

(a) It is easy for humans

(b) It matches electronic ON/OFF states ✅

(c) It uses more digits

(d) It is faster than decimal

8. What is the place value of the rightmost digit in binary?

(a) 2^1

(b) 2^0 ✅

(c) 10^0

(d) 1^2

9. Which of the following represents the correct expansion of a decimal number?

(a) Powers of 2

(b) Powers of 8

(c) Powers of 10 ✅

(d) Powers of 16

10. In decimal to binary conversion, remainders are read:

(a) Top to bottom

(b) Left to right

(c) Bottom to top ✅

(d) Right to left

11. What is the base of the octal number system?

(a) 2

(b) 8 ✅

(c) 10

(d) 16

12. Which digits are used in the octal number system?

(a) 0–1

(b) 0–7 ✅

(c) 0–9

(d) 0–F

13. Each octal digit represents how many binary bits?

(a) 2 bits

(b) 3 bits ✅

(c) 4 bits

(d) 8 bits

14. Why is conversion between binary and octal easy?

(a) Both use same digits

(b) 8 is a power of 2 ✅

(c) Both are base-10

(d) Both use letters

15. What is the octal equivalent of binary 110101011?

(a) 543

(b) 653 ✅

(c) 763

(d) 623

16. When converting binary to octal, digits are grouped in:

(a) Pairs of 2

(b) Groups of 3 ✅

(c) Groups of 4

(d) Groups of 8

17. If a binary number cannot be divided into groups of 3, what should be done?

(a) Remove digits

(b) Add zeros to the right

(c) Add zeros to the left ✅

(d) Multiply by 2

18. In octal system, place values are powers of:

(a) 2

(b) 8 ✅

(c) 10

(d) 16

19. In decimal to octal conversion, the number is divided by:

(a) 2

(b) 4

(c) 8 ✅

(d) 10

20. In decimal to octal conversion, remainders are read:

(a) Top to bottom

(b) Left to right

(c) Bottom to top ✅

(d) Right to left

21. What is the base of the hexadecimal number system?

(a) 2

(b) 8

(c) 10

(d) 16 ✅

22. Which digits are used in the hexadecimal system?

(a) 0–7

(b) 0–9 only

(c) 0–9 and A–F ✅

(d) A–Z

23. In hexadecimal system, the letter 'A' represents which value?

(a) 8

(b) 9

(c) 10 ✅

(d) 15

24. Each hexadecimal digit is equal to how many binary bits?

(a) 2 bits

(b) 3 bits

(c) 4 bits ✅

(d) 8 bits

25. When converting binary to hexadecimal, digits are grouped in:

(a) Groups of 2

(b) Groups of 3

(c) Groups of 4 ✅

(d) Groups of 8

26. What is the maximum value of a 1-byte (8-bit) unsigned integer?

(a) 127

(b) 128

(c) 255 ✅

(d) 256

27. Whole numbers (W) include which of the following?

(a) Negative numbers only

(b) Positive numbers only

(c) Zero and positive integers ✅

(d) Fractions only

28. In a signed integer, the most significant bit (MSB) is used for:

(a) Storing data

(b) Sign representation ✅

(c) Increasing value

(d) Binary conversion

29. What is the maximum positive value of a 1-byte signed integer?

(a) 255

(b) 128

(c) 127 ✅

(d) 256

30. What is the minimum value of an 8-bit signed integer?

(a) -127

(b) -128 ✅

(c) -255

(d) 0

31. Real numbers in computers are also known as:

(a) Integers

(b) Floating-point numbers ✅

(c) Whole numbers

(d) Binary numbers

32. A floating-point number is represented as:

(a) sign × mantissa × 2^exponent ✅

(b) mantissa × sign × exponent

(c) exponent × sign × mantissa

(d) sign × exponent × $mantissa^2$

33. How many bits are used for the exponent in single precision (32-bit)?

(a) 23 bits

(b) 8 bits ✅

(c) 11 bits

(d) 1 bit

34. In single precision (32-bit), how many bits are used for mantissa?

(a) 1 bit

(b) 8 bits

(c) 23 bits ✅

(d) 31 bits

35. What is the total number of bits in double precision format?

(a) 16 bits

(b) 32 bits

(c) 64 bits ✅

(d) 128 bits

36. Which of the following operations are included in binary arithmetic?

(a) Addition, Subtraction, Multiplication, Division ✅

(b) Only Addition

(c) Only Subtraction

(d) Only Multiplication

37. Binary addition uses which digits?

(a) 0–9

(b) 1–10

(c) 0 and 1 ✅

(d) 0–7

38. In binary subtraction, subtraction can be performed using:

(a) Decimal method

(b) Two's complement ✅

(c) Hexadecimal method

(d) Octal method

39. What is the first step in finding the two's complement of a binary number?

(a) Add 1

(b) Divide by 2

(c) Invert all bits ✅

(d) Multiply by 2

40. In binary subtraction using two's complement, what is done with the carry bit?

(a) Added again

(b) Multiplied

(c) Ignored/Discarded ✅

(d) Stored

41. Binary division is similar to which type of division?

(a) Fractional division

(b) Decimal long division ✅

(c) Algebraic division

(d) Matrix division

42. In binary division, which digits are used?

(a) 0–9

(b) 0 and 1 ✅

(c) A–F

(d) 1–8

43. The first step in binary division is:

(a) Add

(b) Compare ✅

(c) Multiply

(d) Subtract

44. If the divisor is less than or equal to the dividend portion, what is done next in binary division?

(a) Multiply

(b) Divide

(c) Subtract ✅

(d) Add

45. How many characters are included in standard ASCII?

(a) 64

(b) 100

(c) 128 ✅

(d) 256

46. Extended ASCII uses how many bits?

(a) 4 bits

(b) 6 bits

(c) 7 bits

(d) 8 bits ✅

47. Unicode can represent approximately how many characters?

(a) 128

(b) 256

(c) Over 1 million ✅

(d) 1024

48. UTF stands for:

(a) Unicode Text Format

(b) Unicode Transformation Format ✅

(c) Universal Text Format

(d) Unique Transfer Format

49. Which encoding scheme is backward compatible with ASCII?

(a) UTF-8 ✅

(b) UTF-16

(c) UTF-32

(d) Extended ASCII

50. Which encoding uses fixed 4 bytes per character?

(a) UTF-8

(b) UTF-16

(c) UTF-32 ✅

(d) ASCII

51. What are the tiny dots that make up an image called?

(a) Bits

(b) Pixels ✅

(c) Frames

(d) Bytes

52. In RGB color model, each pixel color is represented by:

(a) One number

(b) Two numbers

(c) Three numbers (Red, Green, Blue) ✅

(d) Four numbers

53. Which image format supports transparency and lossless compression?

(a) JPEG

(b) PNG ✅

(c) GIF

(d) BMP

54. Which image format is commonly used for animations?

(a) JPEG

(b) PNG

(c) GIF ✅

(d) TIFF

55. What process is used to capture sound waves in audio storage?

(a) Compression

(b) Sampling and quantization ✅

(c) Encoding only

(d) Rendering

56. In audio, sampling refers to:

(a) Converting sound into text

(b) Recording sound at regular intervals ✅

(c) Increasing volume

(d) Compressing files

57. Which audio format is uncompressed and maintains high quality?

(a) MP3

(b) AAC

(c) WAV ✅

(d) MIDI

58. Which video format is widely used and efficiently compresses video?

(a) AVI

(b) MP4 ✅

(c) MKV

(d) MOV

59. Videos are made up of many images called:

(a) Pixels

(b) Frames ✅

(c) Bits

(d) Samples

60. Frame rate is measured in:

(a) Bits per second

(b) Bytes per second

(c) Frames per second (FPS) ✅

(d) Pixels per second

61. What is the base of the decimal number system?

(a) 2

(b) 8

(c) 10 ✅

(d) 16

62. The binary number system uses which digits?

(a) 0–9

(b) 0 and 1 ✅

(c) 0–7

(d) A–F

63. Which number system has base 8?

(a) Decimal

(b) Binary

(c) Octal ✅

(d) Hexadecimal

64. How many digits are used in the hexadecimal number system?

(a) 8

(b) 10

(c) 16 ✅

(d) 2

65. Integers include which of the following?

(a) Only positive numbers

(b) Only negative numbers

(c) Negative, zero, and positive whole numbers ✅

(d) Only fractions

66. Which technique is used to store negative numbers in computers?

(a) One’s complement

(b) Two’s complement ✅

(c) Binary shift

(d) Decimal complement

67. Floating-point numbers are used to represent:

(a) Whole numbers only

(b) Text only

(c) Numbers with fractions/decimal points ✅

(d) Images only

68. ASCII encoding uses how many characters in its basic form?

(a) 64

(b) 100

(c) 128 ✅

(d) 256

69. Unicode is designed to represent:

(a) Only English characters

(b) Only numbers

(c) Characters of all world languages ✅

(d) Only symbols

70. In UTF-32 encoding, each character is stored using:

(a) 1 byte

(b) 2 bytes

(c) 3 bytes

(d) 4 bytes ✅

📘 EXERCISE

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

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

(a) 1101102

(b) 11011 ✅

(c) 110.11 ❌

(d) 1101A

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

(a) 7 bits ✅

(b) 8 bits

(c) 16 bits

(d) 32 bits

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

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

(a) 1 byte

(b) 2 bytes

(c) 4 bytes ✅

(d) 8 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

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

(a) 23 bits

(b) 8 bits ✅

(c) 11 bits

(d) 52 bits

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

(a) 1.4×10^{-45} to 3.4×10^{38} ✅

(b) 1.4×10^{-38} to 3.4×10^{5}

(c) 4.9×10^{-324} to 1.8×10^{308}

(d) 4.9×10^{-308} to 1.8×10^{324}

9. What are the tiny dots that make up an image called?

(a) Pixels ✅

(b) Bits

(c) Bytes

(d) Nodes

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

Short Questions

1. What is the primary purpose of the ASCII encoding scheme?

Ans: The primary purpose of ASCII is to represent characters such as letters, digits, and symbols in a computer using numerical codes so that computers can store and process text.

2. Explain the difference between ASCII and Unicode.

Ans: ASCII uses 7 bits and can represent a limited number of characters mainly in the English language, while Unicode uses more bits and can represent characters from many different languages and symbols worldwide.

3. How does Unicode handle characters from different languages?

Ans: Unicode assigns a unique code point to each character, allowing it to represent text from multiple languages in a single standardized encoding system.

4. What is the range of values for an unsigned 2-byte integer?

Ans: An unsigned 2-byte integer uses 16 bits and can represent values from 0 to 65,535.

5. Explain how a negative integer is represented in binary.

Ans: Negative integers are represented using the two's complement method, where the binary form of a number is inverted (bits flipped) and then 1 is added to the result to obtain its negative representation.

6. What is the benefit of using unsigned integers?

Ans: Unsigned integers allow only non-negative values and therefore provide a larger positive range compared to signed integers using the same number of bits.

7. How does the number of bits affect the range of integer values?

Ans: As the number of bits increases, the range of values also increases because more possible combinations of 0s and 1s can be represented.

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

Ans: Whole numbers are used because quantities like counts, indexes, and sizes cannot be negative, so integer representation is appropriate and efficient.

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

Ans: The range is determined using the IEEE 754 standard, where the sign bit, exponent bits, and mantissa bits together define the smallest and largest representable values.

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

Ans: It is important because floating-point numbers have precision limits and rounding errors, which can affect the accuracy of scientific calculations and results.

Long Questions

🌟 **Q1: Explain how characters are encoded using Unicode. Provide examples of characters from different languages and their corresponding Unicode code points.**

❖ **Answer:**

Unicode is a character encoding standard that assigns a unique number called a code point to every character, symbol, or punctuation mark used in different languages. These code points are written in the form U+XXXX.

In Unicode, each character is first mapped to its code point. Then, this code point is converted into binary form using encoding formats such as UTF-8, UTF-16, or UTF-32. This allows computers to store and process text in a standardized way.

Unicode supports characters from many languages, including English, Urdu, Arabic, and Chinese. For example, the English letter "A" is represented as U+0041, the Urdu letter "ا" is represented as U+0627, the Arabic letter "ب" is U+0628, and the Chinese character "中" is U+4E2D.

Thus, Unicode ensures that each character has a unique identity, allowing text from different languages to be used together without confusion in computers and digital systems.

✦ **Summary:**

Unicode assigns a unique code point to each character, which is then stored in binary form using encoding methods. It supports multiple languages and enables consistent and accurate representation of text across different systems.

🌟 Q2: Describe in detail how integers are stored in computer memory.

❖ **Answer:**

Integers are stored in computer memory using binary representation, which consists of only two digits: 0 and 1. A fixed number of bits (such as 8, 16, 32, or 64 bits) is used to store each integer, depending on the system and data type.

For positive integers, the value is directly converted from decimal to binary and stored in memory. For example, the decimal number 5 is stored as its binary equivalent 00000101 in an 8-bit system.

For negative integers, computers use a method called two's complement representation. In this method, the binary form of the number is first written, then all bits are inverted (0s become 1s and 1s become 0s), and finally 1 is added to the result. This allows the system to represent both positive and negative numbers efficiently.

Integers can be stored as signed or unsigned. Signed integers use one bit (usually the most significant bit) to represent the sign (0 for positive and 1 for negative), while unsigned integers represent only non-negative values and therefore have a larger positive range.

The number of bits used determines the range of values that can be stored. For example, an 8-bit signed integer can store values from -128 to 127, while an 8-bit unsigned integer can store values from 0 to 255.

In memory, integers are stored in binary form in contiguous memory locations, and the system reads and interprets these bits according to the data type and sign format.

✦ **Summary:**

Integers are stored in binary form using a fixed number of bits. Positive integers are stored directly, while negative integers are stored using two's complement representation. The range of values depends on the number of bits and whether the integer is signed or unsigned.

🌟 **Q3: 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 conversion is done by repeatedly dividing the decimal number by 2 and recording the remainders. The binary number is obtained by reading the remainders in reverse order (from bottom to top).

Example (Positive integer):

Convert decimal 13 to binary:

13 ÷ 2 = 6 remainder 1

6 ÷ 2 = 3 remainder 0

3 ÷ 2 = 1 remainder 1

1 ÷ 2 = 0 remainder 1

Reading remainders from bottom to top gives 1101. So, decimal 13 = binary 1101.

Binary to decimal conversion is done by multiplying each binary digit by powers of 2 based on its position (starting from 0 on the right) and then adding the results.

Example:

Convert binary 1101 to decimal:

$(1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$

= 8 + 4 + 0 + 1 = 13

For negative integers, computers use the two's complement method for representation.

Example (Negative integer):

To represent -5 in binary (8-bit system):

1. Write binary of +5 ⟶ 00000101

2. Invert bits ⟶ 11111010
3. Add 1 ⟶ 11111011

So, -5 is represented as 11111011 in two's complement form.

To convert back, the reverse process is applied:

Invert bits, add 1, and then convert to decimal, which gives the magnitude of the negative number.

✦ **Summary:**

Decimal to binary conversion is done using repeated division by 2, while binary to decimal conversion uses positional values (powers of 2). Positive integers are directly converted, whereas negative integers are represented using two's complement method.

🌟 **Q4: Perform the following binary arithmetic operations:**

❖ **Answer:**

a. Multiplication of 101 by 11

Binary multiplication follows the same rules as decimal multiplication, but only 0 and 1 are used.

Step 1: Write the numbers:

101

× 11

Step 2: Multiply 101 by each digit of 11 starting from the right:

Multiply 101 × 1 ⟶ 101

Multiply 101 × 1 ⟶ 101, then shift one place to the left ⟶ 1010

Step 3: Add the partial results:

101

\+ 1010

\---------

1111

Final Answer: 101 × 11 = 1111 (binary)

b. Division of 1100 by 10

Binary division is similar to long division in decimal.

Step 1: Write the division:

110

______

10) 1100

Step 2: Divide step by step:

- Take first two digits: 11
- 10 goes into 11 1 time
- Subtract: 11 − 10 = 1
- Bring down the next digit (0) ⟶ now 10
- 10 goes into 10 ⟶ 1 time

- Subtract: 10 − 10 = 0
- Bring down the last digit (0) ⟶ 0
- 10 goes into 0 ⟶ 0 time

Step 3: Write the quotient:

- Quotient = 110
- Remainder = 0
- Final Answers:
- (a) 101 × 11 = 1111

(b) 1100 ÷ 10 = 110, remainder 0

✦ Summary:

Binary multiplication is done by multiplying each bit and adding shifted results, while binary division is performed using the long division method. Both operations follow simple base-2 arithmetic rules using only 0 and 1.

🌟 Q6: Add the following binary numbers:

a) 101 + 110

Step 1: Write the numbers:

101

 + 110

Step 2: Add from right to left:

- Rightmost: 1 + 0 = 1

- Middle: 0 + 1 = 1
- Leftmost: 1 + 1 = 10 → write 0 and carry 1
- Add carry: 1

Step 3: Final result:

101

+ 110

1011

Answer: 101 + 110 = 1011

b) 1100 + 1011

Step 1: Write the numbers:

1100

+ 1011

Step 2: Add from right to left:

- Rightmost: 0 + 1 = 1
- Next: 0 + 1 = 1
- Next: 1 + 0 = 1
- Leftmost: 1 + 1 = 10 → write 0 and carry 1
- Add carry: 1

Step 3: Final result:

1100

+ 1011

10111

Answer: 1100 + 1011 = 10111

✦ **Summary:**

Binary addition is performed like decimal addition, but whenever the sum equals 2 (10 in binary), we write 0 and carry 1 to the next column.

🌟 **Q7: Convert the following numbers to 4-bit binary and add them:**

We will use 4-bit two's complement representation for negative numbers.

(a) 7 + (−4)

Step 1: Convert 7 to 4-bit binary

7 = 0111

Step 2: Convert −4 to 4-bit (two's complement)

+4 = 0100

Invert bits ⟶ 1011

Add 1 ⟶ 1100

So, −4 = 1100

Step 3: Add the binary numbers

Id="0z6r9y"

0111

+ 1100

10011

In 4-bit addition, we discard the carry beyond 4 bits:

Result = 0011

Final Answer:

7 + (−4) = 0011 (binary) = 3 (decimal)

(b) −5 + 3

Step 1: Convert −5 to 4-bit (two's complement)

+5 = 0101

Invert bits ⟶ 1010

Add 1 ⟶ 1011

So, −5 = 1011

Step 2: Convert 3 to 4-bit binary

3 = 0011

Step 3: Add the binary numbers

Id="b8q7xk"

1011

 + 0011

  1110

Final Answer:

−5 + 3 = 1110 (binary) = −2 (decimal)

✦ **Summary:**

- Positive numbers are directly converted to binary.
- Negative numbers are represented using two's complement.
- After addition, if an extra carry appears, it is discarded in fixed 4-bit representation.

🌟 **Q8: Solve the following binary subtractions:**

(a) 1101_2 – 0100_2

Step:

1101

 - 0100

1001

Answer: 1001_2

(b) 1010_2 – 0011_2

Step:

1010

- 0011

0111

Answer: 0111_2

(c) 1000_2 – 0110_2

Step:

1000

- 0110

0010

Answer: 0010_2

(d) 1110_2 – 0100_2 (Note: 100_2 is written as 0100 for proper 4-bit alignment)

Step:

1110

 - 0100

   1010

Answer: 1010_2

✦ Summary:

Binary subtraction is performed like decimal subtraction using borrowing when needed. Always align numbers with the same number of bits before subtracting.