LDA SCHOOL SYSTEM (BOYS) ALLAMA IQBAL TOWN LAHORE

UNIT # 2 (NUMBER SYSTEMS)

Email: asim.anwar.lda@gmail.com

SHORT QUESTIONS:

<table>
<tr><td>Q1:</td><td>What is meant by numbering systems?</td></tr>
<tr><td>ANS:</td><td>Numbering systems are essential in computing because they form the basis for representing, storing and processing data. Different numbering systems help computers perform tasks like calculations, data storage and data transfer. These systems allow computers to represent various kinds of information, such as text, colors and memory locations.</td></tr>
<tr><td>Q2:</td><td>Define decimal number system?</td></tr>
<tr><td>ANS:</td><td>The decimal number system is a base 10 number system that consists of digits from 0 to 9 and we use it in everyday life. That's why each digit of the number represents a power of 10. In decimal number system the place values starting from the rightmost digits are 10^0,10^1,10^2 and so on. For example the decimal number 578 means:
$5x10^2+7x10^1+8x10^0 = 500+70+8 = 578$</td></tr>
<tr><td>Q3:</td><td>What is binary number system?</td></tr>
<tr><td>ANS:</td><td>The binary number system is a base 2 number system that consists of two digits i.e. 0 and 1. In binary, the place values are arranged from the right to left, starting with 2^0 and ending at 2^n, where each position represents a power of 2. For example the binary number 1011 can be converted to decimal as follows:
$1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = 8 + 0 + 2 + 1 = (11)_{10}$</td></tr>
<tr><td>Q4:</td><td>Write down the method to convert a number from decimal to binary.</td></tr>
<tr><td>ANS:</td><td>1. Divide the decimal number by 2
2. Record the remainder
3. Again divide the number by 2 until the quotient value which left after division becomes 0.
4. Now read the remainder values from bottom to top of the binary number.</td></tr>
<tr><td>Q5:</td><td>Convert 83 to binary.</td></tr>
<tr><td>ANS:</td><td><table>
<tr><td>2</td><td>83</td><td></td></tr>
<tr><td>2</td><td>41</td><td>1</td></tr>
<tr><td>2</td><td>20</td><td>1</td></tr>
<tr><td>2</td><td>10</td><td>0</td></tr>
<tr><td>2</td><td>5</td><td>0</td></tr>
<tr><td>2</td><td>2</td><td>1</td></tr>
<tr><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td>0</td><td>1</td></tr>
</table>
So, $(83)_{10}$ equals to $(01010011)_2$</td></tr>
<tr><td>Q6:</td><td>Define octal system?</td></tr>
<tr><td>ANS:</td><td>Octal is a positional number system with base eight, which implies that a digit to be used ranges from 0 to 7. The last digit is a single digit power of 8 while the other digits are the coefficients. In the decimal system, the place values starting from the 8^0, 8^1,8^2 and so on.</td></tr>
</table>

For example, octal number 157 means, $1x8^2+5x8^1+7x8^0=64+40+7=(111)_{10}$
Each octal digit represents three binary digits.

Q7: **Draw the correspondence table between octal and binary digits.**

ANS:

Octal Digit	Binary Digits
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Q8: **What is the relationship between octal and binary digits?**

ANS: Each octal digit represents three binary digits.

Q9: **Convert $(110101011)_2$ into Octal.**

ANS: First of all divide the number into three digit group starting from right to left.
Now find the each group of three bits in the corresponding octal table/counting .i.e.

110=6
101=5
011=3

So, the binary number 110101011 is equal to 653 in octal number system.

Q10: **Write down the method of conversion from decimal to octal.**

ANS:

- To convert the decimal number into an equivalent octal number, divide the number by 8.
- Write downthe remainder.
- Ather that divide the obtained quotient by 8.
- Continue the division until one of the nubmer results to 0.
- Octal is a base eight number and the octal number is the remainder read it from bottome to top.

Q11: **Convert 83 to octal.**

ANS:

8	83	
8	10	3
8	1	2
	0	1

So, $(83)_{10}$ equal to $(123)_8$

Q12: **Define Hexadecimal number system?**

ANS: The hexadecimal is a base 16 number system with digit number from 0 to 9 and alphabets from A to F, each digit represents 16 to the power of the position of the digit. The letter A to F stand for the numeric value of 10 to 15. The digits in hexadecimal move from right to left in place value that ar $16^0,16^1,16^2$ and so on.

For example 1A3 can be represented in decimal as:
$1x16^2 + Ax16^1 + 3x16^0 = 1x256 + 10x16 + 3x1= 256 + 160 + 3 =(419)_{10}$

Email: asim.anwar.lda@gmail.com

Q13: What is the relationship between binary and hexadecimal digits:

ANS: Every single hexadecimal digit equals four binary digits.

Q14: Draw the hexadecimal and binary digits correspondence table/counting.

ANS:

Hexadecimal Digit	Binary Digits
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Q15: Convert $(1101011010110010)_2$ to hexadecimal.

ANS: Make the groups as four digits starting from right to left.
Now find the each group of four bits in the corresponding hexadecimal table/counting .i.e.

1101=D
0110=6
1011=B
0010=2

So the binary digit **1101011010110010** is equal to D6B2 in hexadecimal system.

Q16: Write down the procedure of converting a number from decimal to hexadecimal.

ANS:

- Convert the decimal number to an absolute value by dividing it by 16.
- Record the quotient and the remainder.
- Continue dividing the quotient by 16 and write down the remainder until the quotient is zero.
- The hexadecimal number is the remainder read it from bottom to top.

Q17: Convert 2297 to hexadecimal.

ANS:

16	2297		
16	143 — 9	→	9
16	8 — 15	→	F
	0 — 8	→	8

↑

So, $(2297)_{10}$ equals to $(8F9)_{16}$

(THREE BASIC RULES FOR CONVERSION)

- **RULE 1:**
  From Decimal Number System to Any other Number System Always use Division Method. (i.e. From Decimal to Binary / Octal / Hexadecimal)

Email: asim.anwar.lda@gmail.com

	• **RULE 2:** From Any other Number System to Decimal Number System Always use Power Method. (i.e. From Binary / Octal / Hexadecimal to Decimal) • **RULE 3:** From Binary to Any other Number System or From Any other Number System Always use Counting Method. (i.e. From Binary to Octal / Hexadecimal or From Octal / Hexadecimal to Binary)
Q18:	**Define whole number (W)?**
ANS:	Whole numbers are a set of non-negative integers. They include zero and all the positive integers. Mathematically, the set of whole numbers is: W={0,1,2,3,……} Usage: Whole numbers are often used to represent quantities that can't be negative. Examples: number of students in a school, a person's age in years etc.
Q19:	**What is the formula to find the maximum value for whole number?**
ANS:	If n is the number of bits, the maximum value that can be represented is 2^n-1. 1 Byte whole number (8 bits): Maximum value = 2^8-1 = 255 2 Byte whole number (16 bits): Maximum value = 2^{16}-1 = 65,535 4 Byte whole numer (32 bits): Maximum value = 2^{32}-1 = 4,294,967,295
Q20:	**Define Integers (Z)?**
ANS:	Integers extends the concept of whole number to include negative numbers. In computer programming, we call them signed integers. The set of integers is represented as: Z={…..,-3,-2,-1,0,1,2,3,…..} To store bothe positive and negative values, one bit is reserved as the sign bit (the most significant bit). If the sign bit is on (1), the value is negative otherwise it is positive.
Q21:	**What is the formula to find maximum value for integer?**
ANS:	As we know one bit is reserve as the sign bit then the bits available to store a value is 2^{n-1}, hence to calculate the maximum value the formula will be 2^{n-1} -1. 1 Byte integer (8 bits): Maximum value = 2^{8-1} - 1 = 127 2 Byte integer (16 bits): Maximum value = 2^{16-1} - 1 = 32,767 4 Byte integer (32 bits): Maximum value = 2^{32-1} - 1 = 2,147,483,647
Q22*:	**What is the range of values for an unsigned 2 byte integer?**
ANS:	The range of values for an unsigned 2 byte integer is 0 to 65,535.
Q23*:	**Explain how a negative integer is represented in binary?**
ANS:	In computing, negative integers are represented by using the two's complement method.
Q24:	**Write down the steps to find two's complement method.**
ANS:	• Find the binary digits and complete the bits. • Invert all the bits (change 0s to 1s and 1s to 0s). • Add 1 to least significant bit (LSB).
Q25:	**Find the two's complement of -5 using 8-bit system.**
ANS:	1. Start with the binary representation of 5 that is 00000101_2. 2. Invert all the bits: 11111010_2. 3. Add 1 : $11111010_2 + 1_2 = 11111011_2$. So, -5 in 8 bit two's complement is 11111011_2. 11111010 +01

Q26*:	**What is the benefit of using unsigned integers?**
ANS:	The main benefit of using integers are that they can store larger positive values (upto double the range of a signed integer of the same size) and their overflow behavior is well defined, unlike the undefined behavior of the signed integers.
Q27*:	**How does the number of bits affect the range of integer values?**
ANS:	Each additional bit used to store an integer **doubles** the number of possible values that can be represented, thereby significantly increasing the range. And it because of the formula 2^n, where n is the number of bits. For example: For 7 bit number, the unique combination will be 2^7-1=127 Whereas for 8 bit number, the unique combination will be 2^8-1=255 So you can observe that with only one bit it doubles the integer range.
Q28:	**What is the formula to find minimum value for integer?**
ANS:	The minimum value is computed using the formula -2^{n-1}, n is the total number of bits. 1 Byte integer (8 bits): minimum value = -2^{8-1} = -128 2 Byte integer (16 bits): minimum value = -2^{16-1} = -32,768 4 Byte integer (32 bits): minimum value = -2^{32-1} = -2,147,483,648
Q29:	**What are the binary addtion rules?**
ANS:	**1.** 0 + 0 = 0 **2.** 0 + 1 = 1 **3.** 1 + 0 = 1 **4.** 1 + 1 = 0 (with a carry of 1 to the next higher bit)
Q30:	**Define subtraction in binary?**
ANS:	In binary arithmetic, subtraction can also be carried out by adding the two's complement or the value of subtrahend to the minuend.
Q31:	**Write down the steps for subtraction in binary.**
ANS:	• Find the two's complement of the subtrahend. ▪ Invert the bits of binary number. ▪ Add 1 to the inverted number. • Add the minuend and the two's complement of the subtrahend. • Discard the carry.
Q32:	**Write down the steps for multiplication in binary.**
ANS:	• Write down the binary number, aligning them by the least significant bit (right most bit). • Multiply each bit of the second number by each bit of the first number, similar to the long multiplication method in decimal system. • Shift the partial results one place to the left for each new row, starting from the second row. • Add all the partial results to get the final product.
Q33:	**Write down the steps for division in binary.**
ANS:	• **Compare:** Compare the divisor with the current portion of the dividend. • **Subtract:** Subtract the divisor from the dividend portion if the divisor is less than or equal to the dividend.

Email: asim.anwar.lda@gmail.com

	• **Shift:** Shift the next binary digit from the dividend down to the remainder. • **Repeat:** Repeat the process until all digits of the dividend have been used.
Q34:	**Why text encoding is important for computers?**
ANS:	Text encoding schemes are essential for representing characters from various languages and symbols in a format that computer can understand and process.
Q35*:	**What is the primary purpose of ASCII encoding scheme?**
ANS:	The primary purpose of the **ASCII** (American Standard Code for Information Interchange) encoding scheme is to provide a standardized system for computers and digital devices to represent text using numerical codes. It enables different computers and devices to exchange text information reliably.
Q36*:	**Explain the difference between ASCII and Unicode.**
ANS:	

Features	ASCII	Unicode
Encoding	Uses 7 or 8 bits per character	Uses variable length encodings (UTF-7, UTF-8, UTF-16, UTF-32) which use 7, 8 , 16 or 32 bits
Number of Characters	128	Over 144,000
Languages Supported	Primarily English alphabets, digits and symbols	Virtually all languages, scripts and symbols
Memory Usage	Use less memory per character	Uses more memory per character than ASCII
Relationship	A subset of Unicode	A superset of ASCII

Q37*:	**How does unicode handles characters from different languages?**
ANS:	Unicode handles characters from different languages by assigning a unique numerical code point to every character, symbol, and emoji, regardless of its script. This allows computers to represent and display text from any language by using a standard mapping, which is then converted into bytes for storage by an encoding scheme like UTF-8 etc.

LONG QUESTIONS:

Q1:	**Explain how characters are encoded using Unicode. Provide examples of characters from different languages and their corresponding Unicode code points.**
ANS:	Unicode is an international character encoding standard designed to represent text from virtually all of the world's writing systems. It assigns a unique numeric value, known as a code point, to each character, regardless of the platform, program, or language. This provides a universal way to identify and process characters. **How Unicode Encoding Works:**

Email: asim.anwar.lda@gmail.com

	Code Points: Unicode assigns a distinct code point (a number) to each character. These code points are typically represented in hexadecimal notation, prefixed with "U+". For example, the Latin capital letter 'A' is U+0041. **Character Encoding Forms:** While Unicode defines the code points, it doesn't specify how these code points are stored as bytes in computer memory. This is where Unicode Transformation Formats (UTFs) come in. The most common UTF encodings are: **UTF-8:** A variable-width encoding that uses one to four bytes per character. It is backward compatible with ASCII, meaning ASCII characters are represented by a single byte in UTF-8. It's widely used on the web and in many operating systems. **Example:** The English letter 'A' in UTF-8, is represented as, U+0041, is 01000001 in the binary format and occupies 8 bits or 1 byte. **Example:** The Urder letter 'ب' in UTF-8, is represented as, U+0628, is 11011000 10101000 in the binary format, means it takes 2 bytes. **UTF-16:** A variable-width encoding that uses two or four bytes per character. It is commonly used in internal string representations in some programming languages and operating systems. **Example:** The English letter 'A' in UTF-16 is equal to 00000000 01000001 in binary or 65 in decimal (2 bytes). **Example:** The Urder letter 'ب' in UTF-16 is represented in binary as 00000110 00101000 which occupies 2 bytes of memory. **UTF-32:** A fixed-width encoding that uses four bytes per character. While simpler in its fixed-width nature, it can be less space-efficient for text primarily composed of characters that fit into fewer bytes. **Example:** The English letter 'A' in UTF-32 is represented in binary as 00000000 00000000 00000000 01000001 which occupies 4 bytes of memory. **Example:** The Urdu letter 'ب' in UTF-32 is represented in binary as 00000000 00000000 00000110 00101000 which occupies 4 bytes of memory. **Examples of Characters and their Unicode Code Points:** Here are examples of characters from different languages and their corresponding Unicode code points: These examples demonstrate how Unicode provides a consistent and universal way to identify and represent characters from diverse linguistic and symbolic systems, enabling seamless text processing across different platforms and applications.
Q2:	**Describe in detail how integers are stored in computer memory?**

ANS:

Integers are stored in computer memory as a sequence of bits, using binary numbers, where each bit is either a 0 or a 1. The size of this sequence, determined by the data type (e.g., 8-bit, 16-bit, 32-bit, 64-bit), defines the range of numbers that can be stored. To represent both positive and negative numbers, computers primarily use the two's complement method, where the leftmost bit indicates the sign (0 for positive, 1 for negative).

Key concepts

- **Binary Representation**: All data in a computer, including integers, is stored as a series of 0s and 1s. Each 0 or 1 is called a bit.
- **Fixed Size**: The number of bits used to store an integer is fixed for its data type. Common sizes are 8, 16, 32, and 64 bits, which correspond to 1, 2, 4, and 8 bytes, respectively. These data types determine the range of values it can represent.
    - **8-bit (1 byte):** range 0 to 255 (unsigned), or -128 to 127 (signed)
    - **16-bit (2 bytes):** range 0 to 65,535 (unsigned), or -32,768 to 32,767 (signed).
    - **32-bit (4 bytes):** range 0 to (4,294,967,295) 4 billion (unsigned), or (-2,147,483,648 to 2,147,483,647) approx. -2.1 billion to +2.1 billion (signed).
    - **64-bit (8 bytes):** used for very large integers with significantly larger ranges.
- **Signed vs. Unsigned**:
    - **Unsigned integers** can only represent non-negative numbers (0 and up).
    - **Signed integers** can represent both positive and negative numbers. The most significant bit (the leftmost one) is used as a sign bit.

Storing signed integers (Two's Complement)

This is the most common method for storing signed integers.

- **Positive numbers**: The leftmost bit is a 0, followed by the binary representation of the number. For example, the 8-bit binary for 5 would be 00000101.
- **Negative numbers**: To find the two's complement representation of a negative number:
    1. Take the binary representation of the positive version of the number.
    2. Invert all the bits (0 becomes 1, and 1 becomes 0). This is called the one's complement.
    3. Add 1 to the result.
- **Example**: Let's find the two's complement for −5 using 8 bits:
    1. Positive 5 in 8 bits is 00000101.
    2. Invert the bits (one's complement): 11111010.
    3. Add 1: 11111010 + 1 = 11111011. So, 11111011 is the two's complement representation of -5.

Email: asim.anwar.lda@gmail.com

Why two's complement?

- **Efficiency**: This system allows the processor to use the same circuitry for both addition and subtraction. Subtraction is performed by adding the two's complement of the subtrahend to the minuend.
- **No separate zero**: It has a single, unambiguous representation for zero, unlike some other methods.

Q3: **Explain the process of converting a decimal integer to its binary representation and vice versa. Include examples of both positive and negative integers.**

ANS:

Decimal to Binary Conversion:

Positive Integers (Division Method):

- Divide the decimal number by 2 and note the remainder (0 or 1).
- Repeat the process with the resulting quotient until the quotient becomes 0.
- Write the remainders in reverse order (from bottom to top) to get the binary number.

Example: Convert 13 to binary

13 ÷ 2 = 6, remainder **1**

6 ÷ 2 = 3, remainder **0**

3 ÷ 2 = 1, remainder **1**

1 ÷ 2 = 0, remainder **1**

Reading the remainders from bottom to top: **1101**

Negative Integers (Two's Complement Method):

- Convert the absolute value of the number to binary.
- Ensure the binary number has a fixed number of bits (e.g., 8 bits).
- Invert all the bits (change 0s to 1s, 1s to 0s) to get the one's complement.
- Add 1 to the one's complement to get the two's complement representation.

Example: Convert -6 to binary (using 8 bits)

Absolute value 6 in binary: 0000 0110

Invert the bits: 1111 1001 (one's complement)

Add 1: 1111 1010 (two's complement)

-6 in binary is **1111 1010**.

Binary to Decimal Conversion:

Positive Integers (Positional Notation Method):

- Multiply each binary digit by its corresponding power of 2, starting from 2^0 for the rightmost digit and increasing the power by one for each position to the left.
- Sum all the products to get the decimal equivalent.

Example: Convert 1101 to decimal

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

$= (1 \times 8) + (1 \times 4) + (0 \times 2) + (1 \times 1)$

$= 8 + 4 + 0 + 1 =$ **13**

Negative Integers (Two's Complement Method):

- Identify that the most significant bit (leftmost) is 1, indicating a negative

Email: asim.anwar.lda@gmail.com

	number. • Invert all bits of the binary number. • Add 1 to the result. • Convert this new positive binary number to decimal and add a negative sign. **Example: Convert 1111 1010 to decimal** **Invert the bits:** 0000 0101 **Add 1:** 0000 0110 **Convert:** 0000 0110 to decimal (as above): 6 **Add the negative sign: -6**
Q4(a):	**Multiplication of 1 0 1 by 1 1.**
ANS:	1 0 1 x 1 1 ——— 1 0 1 1 0 1 x ——— 1 1 1 1
Q4(b):	**Division of 1 1 0 0 by 1 0.**
ANS:	10) 1 1 0 0 (110 - 1 0 ——— 1 0 - 1 0 ——— 0
Q5(a):	**Add 1 0 1 to 1 1 0.**
ANS:	1 0 1 + 1 1 0 ——— 1 0 1 1
Q5(b):	**Add 1 1 0 0 to 1 0 1 1.**
ANS:	1 1 0 0 + 1 0 1 1 ——— 1 0 1 1 1
Q6(a):	**Convert the numbers into 4-bit binary and add them: 7 + (- 4)**
ANS:	4-bit binary representation of 7 is **0111** 4-bit binary representation of 4 is **0100** • Invert the bits: 0100 to 1011 • Add 1 to the result: 1011 + 1 = 1100 Now, 4-bit binary representation of – 4 is **1100** **Addition:** 0111 + 1100 ------- **1**0011

Email: asim.anwar.lda@gmail.com

<table>
<tr><td></td><td>Since we are working with 4-bit binary, so we discard the fifth bit (the carry bit). So the final 4-bit result is 0011.</td></tr>
<tr><td>Q6(b):</td><td>Convert the numbers into 4-bit binary and add them: -5 + 3</td></tr>
<tr><td>ANS:</td><td>Decimal 5 in 4-bit representation is 0101
• Invert the bits: 0101 to 1010
• Add 1 to result: 1011
Decimal - 5 in 4-bit representation is 1011
Decimal 3 in 4-bit representation is 0011
Addition:
  1011
+ 0011

  1110
Sum of -5 +3 in 4-bit binary result is 1110</td></tr>
<tr><td>Q7(a):</td><td>Solve: 1101_2 - 0100_2</td></tr>
<tr><td>ANS:</td><td> 1 1 0 1
- 0 1 0 0
  1 0 0 1</td></tr>
<tr><td>Q7(b):</td><td>Solve: 1010_2 - 0011_2</td></tr>
<tr><td>ANS:</td><td>• Find the two's complement of the subtrahend (0011), which is 1101.
• Add the two's complement to the minuend: 1010+1101=10111
• Discard the overflow bit (the leading '1'), resulting in 0111.</td></tr>
<tr><td>Q7(c):</td><td>Solve: 1000_2 - 0110_2</td></tr>
<tr><td>ANS:</td><td> 1 0 0 0
- 0 1 1 0
  0 0 1 0</td></tr>
<tr><td>Q7(d):</td><td>Solve: 1110_2 - 100_2</td></tr>
<tr><td>ANS:</td><td> 1 1 1 0
- 0 1 0 0
  1 0 1 0</td></tr>
</table>