UNIT 02

Number Systems

Short Introduction of Unit

Understanding number systems is fundamental in computer science and digital electronics. This chapter will delve into various numbering systems, their applications, and how they are used in computers.

Q.1 Explain Numbering System with its types in details. 09502001

Ans. 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. Here is a description of a few numbering systems that are used most frequently:

1. Decimal System

The decimal numbering system is a base-10 number system that uses digits from 0 to 9 and we use it in our daily life. Each digit represents a power of 10 in decimal, the place values from right to left are 10^0, 10^1 ,10^2, and so on. For example, The decimal number 523 means:

$5x10^2+2x10^1+3x10^0=500+20+3=523$

Did You Know?
The concept of **zero** as both a **number** and a **placeholder** was introduced by **Indian Mathematicians**.
Brahmagupta (598 CE - 668 CE)

$\therefore 10^0 = 1$

2. Binary System

The binary numbering system is a base-2 number system that uses digits from 0 to 1. Each digit represents a power of 2 In binary, the place values from right to left are 2^0,2^1,2^2, and so on. For example. The binary 1011 means:

$1\times2^3+0\times2^2+1\times2^1+1\times2^0=8+0+2+1=11_{10}$

Conversion from Decimal to Binary

The following algorithm converts a decimal number to binary.

- Divide the decimal number by 2.
- Record the remainder.
- Repeat the division by 2 until the quotient is 0.
- The binary number is the remainders read from bottom to top.

Example: Convert 83 to binary

83 / 2 =41 remainder 1
41 / 2 =20 remainder1
20 / 2 =10 remainder 0
10 / 2 =5 remainder 0
5 / 2 =2 remainder 1
2 / 2 =1 remainder 0
1 / 2 =0 remainder 1

OR

2	83
2	41 – 1
2	20 – 1
2	10 – 0
2	5 – 0
2	2 – 1
2	1 – 0
	0 – 1

Decimal to Binary Conversion

3. Octal System

The octal numbering system is a base-8 number system that uses digits from 0 to 7 Each digit Represents a power of 8. In octal, the place values from right to left are 8^0,8^1,8^2, and so on. For example, the octal number 157 means:

$1\times8^2+5\times8^1+7\times8^0=64+40+7=111_{10}$

Did you know?
The octal system was more common in early computing systems like the PDP-8.

Each octal digit represents three binary digits (bits)because the octal system is base-8, and the binary system is base-2. This relationship arises from the fact that 8 is a power of $2(8=2^3)$.So, each over all digit can be precisely represented by three binary digits (bits).

Example:

Consider the 9- bit binary number 110101011. This number can be divided into group of three.

Bits from right to left:

110 101 011

Each group of three bits corresponds to a single octal digit:

110=6

101=5

011=3

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

Correspondence between Octal and Binary Digits

So, the binary numbers 110101011 is equivalent to the octal number 653.

Conversion from Decimal to Octal

The following algorithm converts a decimal number to octal.

1. Divide the decimal number by 8.
2. Write down the remainder.
3. Divide the quotient by 8.
4. Repeat until the quotient is 0.
5. The octal number is the remainders read from bottom to top.

Example: Convert 83 to octal

- 83/ 8=10 remainder 3
- 10/8=1 remainder 2
- 1/ 8=0 remainder 1

OR

8	83
8	10 – 3
8	1 – 2
	0 – 1

Decimal to Octal

So, $83_8 = 123_{10}$

Q.2 Explain Hexa-decimal number system with suitable examples. 09502002

Ans. The hexadecimal numbering system is a base-16 number system that uses digits from 0 to 9 and letters from A to F Each digit represents a power of 16. The letters A to F represent the values 10 to 15 respectively as shown in Table. In hexadecimal, the place values from right to left are 16^0,16^1,16^2, and so on. For Example, the hexadecimal number 1A3 means:

$1\times16^2+A\times16^1+3\times16^0=1\times256+10\times16+3\times1=256+160+3=419_{10}$

Each hexadecimal digit represents four binary digits (bits) because the hexadecimal system is base-16 and the binary system is base-2. This relationship arises from the fact that 16 is a power of 2 ($16=2^4$). This means that any value from 0 to 15 in hexadecimal can be converted into a 4-bit binary number. This relationship makes conversion between binary and hexadecimal. Table shows the correspondence between hexadecimal and binary digits:

Example: Consider the 16-bit binary number 1101011010110010.This number can be divided into group of four bits from right to left:

1101 0110 1011 0010

Hexadecimal	Binary
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

Correspondence between Hexadecimal and Binary Digits

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

Converting Decimal to Hexadecimal

The following algorithm converts a decimal number to hexadecimal:

1. Convert the decimal number to an absolute value by dividing it by 16.
2. Record the quotient and the remainder.
3. Continue dividing the quotient by 16 and write down the remainder until the quotient is zero.
4. The hexadecimal number, as you might have guessed, is the remainder read from bottom to top.

Example: Convert 2297 to hexadecimal

2297 /16=143 remainder 9
143/16=8 remainder F
8 /16=0 remainder 8

OR

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

Decimal to Hexadecimal

Q.3 Describe in detail how integers are stored in computer memory. 09502003

Ans: Computers are amazing machines that can process and store a lot of information. In the following section we will see numeric data representation.

Binary Encoding of Integers(Z) and Real Numbers(R)

When we store data in computers especially numbers, it's important to understand how they are represented and stored in memory. Let's explore how different size of integer values are stored in 1, 2 and 4 bytes and how we can store positive and negative integers.

Whole Numbers (W) and Integers(Z)

Integers, also known as whole numbers, are important elements in both mathematics and computer science. Knowledge of these concepts is important for primary computations, solving problems through programming, working with data and designing algorithms.

Whole Number (W)

Whole numbers are the set of non-negative numbers. They include zero and all the positive integers. Mathematically, the set of whole numbers is:

$$W=\{0, 1, 2, 3, \ldots \ldots\}$$

In computing, whole numbers are often used to represent quantities that can't be negative. Examples include the number of students in a school, a person's age in years, and grades, provided there are no negative figures such as a credit point balances.

1-byte integer has 8 bits to store values. If all 8 bits are **ON,** it represents the maximum value, 11111111_2 which is 255_{10}. If all bits are **OFF**, it represents the minimum value 00000000_2, which is 0_{10}. If n is the number of bits, the maximum value we can represent is 2^n -1. For example,

- 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 number (32 bits): Maximum value=2^{32}-1=4,294,967,295

Integers(Z)

Integers extend the concept of whole numbers to include negative numbers. In the world of computer programming. We call them signed integers. The set of integers is represented as:

$$Z= \{\ldots\ldots -3, -2, -1, 0, 1, 2, 3\ldots\ldots\}$$

To store both positive and negative values, one bit is reserved as the sign bit (the most significant bit).

Negative Values and Two's Complement

In order to store negative values. Computers use a method called two's complement. To find the two's complement of a binary number, follow these steps:

1. Invert all the bits (change 0s to 1s and 1s to 0s).
2. Add 1 to the least significant bit(LSB).

Example: Let's convert the decimal number -5 to an 8 bit binary number:

1. Start with the binary representation of 5:00000101_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.

Did you know?

The smallest positive number representable in single precision is approximately $1.4\text{x}10^{-45}$ and in double precision, it is approximately $4.9\text{x}10^{-324}$

Minimum Integer Value

For 8-bit integer, we switch on the sign bit for the negative value and make all bit ON which results 11111111_2. Except the first bit, we take two's complement and get 10000000_2. Which is 128_{10}. So the minimum value in 1- byte integer is -128 , i.e., -2^7 . So, the minimum value is computed using the formula -2^n, where n is the total number of bits.

- **2-Byte Integer (16 bits):** Minimum value= -2^{15}= -32,768
- **4-Byte Integer (32 bits):** Minimum value= -2^{31}-1= -2,147,483,648

Q.4 How to store real or decimal values in computer memory? 09502004

Ans: In computers, real values (also known as floating- point numbers) are used to represent numbers that have fractions or decimal points.

Understanding Floating Point Representation

Floating-point numbers (real values) are represented in a way similar to scientific notation given below:

Floating –point number =sign x mantissa x $2^{exponent}$. According to the above formula, 5.75 is represented as 1.4375 x 2.

Step for Conversion

1. **Identify the Fractional Part:** Extract the fractional part of the decimal number, For example. If the number is 4.625, the integral part is 4 and the fractional part is 0.625.
2. **Convert the Fractional part to Binary:** Multiply the fractional part by 2 and record the integer part of the result. Continue this process with the new fractional part until it becomes 0 or until you reach the desired precision.

Example: Converting 0.375 to Binary

1. **Identify the Fractional Part:** Fractional part:0.375
2. **Convert the Fractional Part:** 0.375 to Binary.

   0.375 ×2=0.75 (Integer part:0)
   0.75×2=1.5 (Integer part:1)
   0.5×2=1.0 (Integer part:1)

The integer parts recorded are 0,1,1

3. **Combine the Results:-** Combine the binary representations of the integer parts from top to bottom:

   $0.375_{10}=0.011_2$

In computer science, representing real numbers in binary form is essential for efficient computation and storage. This process involves converting both the integer (decimal) and fractional parts of as number into binary. The two most commonly used standard are **"single precision (32-bits) and Double Precision (64-bit)"**

Q.5 Differentiate between Single and Double Precision. 09502005

Ans: Single Precision (32-bit)

In this standard, 4 bytes (32 bits) are used where 1 bit is for the sign, next 8 bits for the exponent and the last 23 bits for the mantissa.

- The exponent ranges from -126 to +127.
- The approximate range of values: $1.4x10^{-45}$ to $3.4x10^{38}$.

Did You Know?

1 Byte (B) =8 Bits
I Kilobyte (KB)=1024 Bytes
1 Megabyte (MB) =1024 Kilobyte
1 Gigabyte (GB) =1024 Megabytes
1 Terabyte (TB) =1024 Gigabytes
1 Petabytes (PB) =1024 Terabyte
1 Exabyte (EB) =1024 Petabytes
1 Zettabyte (ZB) =1024 Exabytes
1 Yottabyte (YB)= 1024 Zettabytes

Value	Representation	Sign Bit	Expenent (8 bits)	Mantissa (23 bits)
Grouping		1 bit	8 bits	23 bits
5.75	$1.4375x2^2$	0	10000001	10111000000000000000000
-5.75	$-1.4375x2^2$	1	10000001	10111000000000000000000
0.15625	$1.25x2^{-3}$	0	01111101	01000000000000000000000
-0.15625	$-1.25x2^{-3}$	1	01111101	01000000000000000000000

32-bit Floating point Representation

Explanation:

Table illustrates how 32 bits floating point values are represented in binary form. Each floating point value is broken down into three main components. The sign bit, the exponent, and mantissa.

Grouping: This row explains the bit allocation for the 32-bit floating point format: 1 bit for the sign, 8 bits for exponent, and 23 bits for the mantissa.

1. **5.75:** Representation: 1.4375×2^2- sign Bit: 0(positive)-Exponent: 2+127=129, which is 10000001_2 Mantissa: the binary representation of 0.4375 is $10111000000000000000000_2$
2. **-5.75:** Representation: -1.4375×2^2-Sign Bit:1(negative) -Exponent:2 +127=129. which is 10000001_2 Mantissa: The binary representation of 0.4375 is $10111000000000000000000_2$
3. **0.15625:** Representation:1.25×2^{-3} -Sign bit:0 (positive)-Exponent: -3+127=124, which is 01111101_2 Mantissa: The binary representation of 0.25 is $01000000000000000000000_2$
4. **-0.15625:** -Representation: -1.25×2^{-3} Sign bit:1 (negative) Exponent: -3+127=124, Which is 01111101_2 Mantissa: The binary representation of 0.25 is $01000000000000000000000_2$ This breakdown help illustrates how floating-point values are stored and manipulated in computer systems.

Double Precision (64-bit)

In double precision, the exponents is represented using 11 bits. The exponent is stored in a biased form with a bias of 1023. The range of the actual exponent values can be determined as follows:

- **Bias:**1023
- **Exponent range:** The actual exponent values range from -1022 to +1023. Therefore, the smallest and largest possible exponent values in double precision are:
- **Minimum exponent:** -1022
- **Maximum exponent:** +1023

Q.6 What are the basic Binary Arithmetic Operations? Explain in details. 09502006

Ans. Binary Arithmetic refers to the operations of addition, subtraction, multiplication, and division performed on binary numbers. Binary numbers are the basis of all operations in digital computers. Binary arithmetic operations are similar to decimal operations but follow binary rules. Here's a brief overview of basic operations:

1. Addition

Binary addition uses only two digits: 0 and 1 Here, we will learn how to add binary numbers and also how to handle the addition of negative binary numbers.

Binary Addition Rules

Binary addition follows these simple rules:

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)

Example of Binary Addition:

Example:

$$\begin{array}{r} 1101 \\ +1011 \\ \hline 011000 = 24 \text{ (Decimal)} \end{array}$$

In this example:

- 1+1=0 (carry1)
- 0+1+1 (carry)= 0 (carry 1)
- 1+0+1(carry) =0 (carry 1)
- 1+1+1 (carry)= 1 (carry 1)

2. Subtraction

In binary arithmetic, subtraction can be performed by adding the two's complement of the subtrahend (the number being subtracted) to the minuend (the number from which another number is subtracted).

Example: Subtract 6 from 9 in Binary

Minuend $=9_{10}=1001_2$

Subtrahend$=6_{10}=0110_2$

Step 1: Find the Two's Complement of the Subtrahend

- Invert the bits of 0110_2.
  Inversion: 1001_2
- Add 1 to the inverted number:
  $1001_2+1_2=1010_2=-6_{10}$

Step 2: Add the Minuend and the Two's Complement of the subtrahend

$1001_2+1010_2=10011_2$

Step 3: Discard the Carry Out

$10011_2 \rightarrow$ Discard carry $\rightarrow 0011_2=3_{10}$

So, 9-6=3.

3. Multiplication

Binary numbers are base-2 numbers. Consisting of only 0s and 1s. Multiplying binary numbers following similar principles to multiplying decimal numbers, but with simpler rules. Here, we will learn how to multiply binary numbers with example and practice questions.

Step to Multiply Binary Numbers

1. Write down the binary numbers, aligning them by the least significant bit (rightmost bit).
2. Multiply each bit of the second numbers by each bit of the first numbers, similar to the long multiplication method in decimal.
3. Shift the partial results one place to the left for each new row, starting from the second row.
4. Add all the partial results to get the final product.

Example

Let's multiply two binary numbers: 101_2 and 11_2.

$$\begin{array}{rr} & 1\ 0\ 1 \\ \times & 1\ 1 \\ \hline & 1\ 0\ 1 \quad (\text{This is } 101_2 \times 1_2) \\ & 1\ 0\ 1 \times\ 0 \quad (\text{This is } 101_2 \times 1_2, \text{ Shifted left}) \\ \hline 1\ 1\ 1\ 1 & \end{array}$$

So, $101_2 \times 11_2 = 1111_2$

4. Division

Binary division is similar to decimal division but only involves two digits, 0 and 1. Binary division includes steps like comparing, subtracting, and shifting. Akin to long division in the decimal system.

Steps of Binary Division

1. **Compare:** Compare the divisor with the current dividend portion.
2. **Subtract:** Subtract the divisor from the dividend portion if the divisor less than or equal to the dividend.
3. **Shift:** Shift the next binary digit from the dividend down to the remainder.
4. **Repeat:** Repeat the process until all digits of the dividend have been used.

Example

Divide 1100_2 by 10_2

$$
\begin{array}{r}
110 \\
10\overline{)1100} \\
-10 \\
\hline
10 \\
-10 \\
\hline
0
\end{array}
$$

(Step 1: Compare 10 with first two digits 11, Subtract 10 from 11)
(Step 2: Bring down the next digit 0)
(Step 3: Compare 10 with 10, subtract 10 from 10)
(Step 4: Bring down the next digit 0, no more digits left)

Result: $1100_2 / 10_2 = 110_2$

Q.7 Describe the history and development of ASCII. Explain its limitations and why Unicode was developed. Discuss how Unicode extends the capabilities of ASCII. 09502007

Ans. Text encoding schemes are essential for representing characters from various languages and symbols in a format that computers can understand and process. Here are some of the most common text encoding schemes used in computers:

1. ASCII

ASCII stands for American Standard Code for information Interchange. It is character encoding standard used to represent text in computers and other devices that use text. Each letter, digit, or symbol is assigned a unique number between 0 and 127, as shown in Table. The ASCII code for 'P' is 80.

The ASCII code for 'a' is 97.
The ASCII code for 'k' is 107.
The ASCII code for 'i' is 105.
The ASCII code for 's' is 115.
The ASCII code for 't' is 116.

The ASCII code is a numerical representation of characters in computer-based system, particularly for alphabetic characters. For example, the ASCII code of the character 'n' is 110.

Extended ASCII

While the standard ASCII table includes 128 characters, there is an extended version that includes 256 characters. This extended ASCII uses 8 bits and includes additional symbols, accented letters, and other characters. However, the original 128 characters are the most commonly used and are the basis for text representation in computers.

Character	ASCII Code	Character	ASCII Code
SP (space)	32	!	33
"	34	#	35
$	36	%	37
&	38	'	39
(	40	)	41
*	42	+	43
,	44	-	45

.	46	/	47
0	48	1	49
2	50	3	51
4	52	5	53
6	54	7	55
8	56	9	57
:	58	;	59
<	60	=	61
>	62	?	63
@	64	A	65
B	66	C	67
D	68	E	69
F	70	G	71
H	72	I	73
J	74	K	75
L	76	M	77
N	78	O	79
P	80	Q	81
R	82	S	83
T	84	U	85
V	86	W	87
X	88	Y	89
Z	90	[	91
\	92	]	93
^	94	-	95
?	96	a	97
b	98	c	99
d	100	e	101
f	102	g	103
h	104	i	105
j	106	k	107
l	108	m	109
n	110	o	111
p	112	q	113
r	114	s	115
t	116	u	117
v	118	w	119
x	120	y	121
z	122	{	123
|	124	}	125
~	126	DEL (delete)	127

ASCII Table

2. Unicode

Unicode is a character encoding standard that aims to cover all the characters used in the world's writing systems. Unlike ASCII, which only uses 7 bits and can represent 128 characters. Unicode

can represent over a million characters using different encoding forms like UTF-8, UTF-16, and UTF-32 UTF stands for **Unicode Transformation Format.**

Q.8 Explain how characters are encoded using Unicode. Provide examples of characters from different languages and their corresponding Unicode points. Discuss the importance of Unicode in global communication and software development. 09502008

Ans. UTF-8

It is a variable-length encoding scheme, which means it can use different numbers of bytes (from1 to 4) represent a character. UTF-8 is backward compatible with ASCII. It means that UTF-8 can understand and use the older ASCII encoding scheme without any problems. So, if we have a text file written in ASCII, it will work perfectly fine with UTF-8. It's like being able to read both old and new books.

Example: The letters '**A**' is Unicode (which is U+0041) is represented as 01000001 in binary format and occupies 8 bits or 1 byte.

Let's look at how Urdu letters are represented in UTF-8:

Example: The letter '**ب**' (which is U+0628 in Unicode) is represented as 1101100010101000 in binary (2 bytes).

UTF-16

UFT-16 is another variable-length encoding scheme, but it uses 2 or 4 bytes for each character. Unlike UTF-8, it is not backward compatible with ASCII

Example: The letter A is represented as 00000000 01000001 in binary (2 bytes).

For Urdu

Example: The Urdu letter '**ب**' is represented as 00000110 00101000 in binary (2 bytes).

UTF-32

UFT-32 is a fixed -length encoding scheme. Every character, no matter what, is represented using 4 bytes. This makes it very simple, but at the same time it may look a little complicated when it comes to space usage.

Example: The letter A is represented as 00000000 00000000 00000000 01000001 in binary (4 bytes).

Q.9 Describe the concept of Storing Images, Audio and Video in Computers. 09502009

Ans: Have you ever wondered how your favorite photos, songs, and movies are stored on your computer or phone? Let's dive into the fascinating world of digital storage to understand how computers manage these different types of files.

1. Storing Images

Images are made up of tiny dots called **pixels.** Each pixel has a color, and the combination of all these pixels forms the complete picture. Computers store images using numbers to represent these colors.

Did You Know?

The first hard drive, created **IBM** in **1956,** weighed over a ton, and could only store 5 megabytes of data.

- **Color Representation:** In a color image each pixel's color can be represented by three numbers: Red, Green, and Blue (RGB). Each of these numbers typically ranges from 0 to 255.- For Example, a pixel with RGB values (255,0,0) will be bright red.
- **Image File Formats: JPEG (Joint Photographic Expert Group):** Common format for photos. It compresses the image to save space but might lose some quality.
- **PNG (Portable Network Graphics):** Supports transparency and maintains high quality without losing data.
- **GIF (Graphics Interchange Format):** Used for simple animations and images with few colors.

2. Storing Audio

Audio files are stored by capturing sound waves and converting them into digital data. This process involves sampling and quantization.

- **Sampling:** Recording the sound wave at regular intervals. The number of samples per second is called the **sampling rate.** Higher sampling rates means better quality.
- **Quantization:** Converting each sample into a number. More bits per sample mean more accurate sound representation.

Audio file Formats

- **MP3:** A common format that compresses audio to save space but may lose some quality.
- **WAV (Wave Audio File Format):** Uncompressed format that maintains high quality.
- **AAC (Advanced Audio Coding):** Used by many streaming services for high- quality audio with efficient compression.

Examples: To illustrate how these units are used, consider the following example:

- An image file might have a size of 500 KB (Kilobytes).
- A music file might be around 5MB (Megabytes).
- A full-length HD movie could be approximately 2 GB (Gigabytes).
- A large external hard drive could have a capacity of 1 TB (Terabytes).

3. Storing Video

Videos are made up of many images shown rapidly one after another, along with audio. Each image in a video is called a **frame.**

Frames and Frame Rate

- **Frame Rate:** The number of frames shown per second, measured in Frames Per Second (FPS). Common frame rates are 24 fps (use in movies) and 30 fps (used in TV). Higher frame rates result in smoother motion in videos.

Video File Formats

- **MP4:** Widely used format that efficiently compresses video while maintaining quality.
- **AVI:** Older format that may result in large file sizes.
- **MKV:** Supports high- quality video and multiple audio tracks/ subtitles.

How Computers Store These Files?

All these files (images, audio, and video) are **Binary data,** which means they are represented by sequences of 0s and 1s.

Storage Devices

- **Hard Drives (HDD):** Uses spinning disks to read/write data. They offer large storage capacities.
- **Solid State Drives (SSD):** Use flash memory for faster access times and better performance.
- **Cloud Storage:** Storage files on remote servers accessible via internet, providing flexibility and backup.

Topic Wise Short Questions (Additional)

Number Representation & Conversion

Q.1 Store the word "Phone" in computer memory starting from address 7003 where each letter needs one byte to store in the memory. 09502010

Ans:

Human's View about Memory	Code in Decimal	Code in Binary	Address
'P'	80	01010000	7003
'h'	104	01101000	7004
'o'	111	01101111	7005
'n'	110	01101110	7006
'e'	101	01100101	7007

Q.2 What is number system? 09502011

Ans: A number system is the system for representation of numeric data. A Number system is defined as a set of values used to represent different quantities. We all are familiar with decimal number system where each number consists of digits from 0 to 9. In a computer system, other number systems are also used. e.g. Binary, Hexadecimal etc.

Q.3 What do you mean by conversion of number system? 09502012

Ans. A process to convert one number system to another number system is called conversion of number system. For example, converting a decimal number to binary number.

Q.4 How data is represented in computer memory? 09502013

Ans: Digital computers store data in binary form. It means that whether it is a text, picture, movie or some application, it is stored in computer's memory in the form of 0s and 1s.

ASCII

Q.5 What is ASCII? 09502014

Ans. ASCII (American Standard Code for Information Interchange) is one such coding scheme published by **ISO (International Standards Organization)**. It is 7-bit coding scheme. The codes are assigned to various characters. Most computers also use 8-bit ASCII code.

Example: A = 65, B = 66 and a = 97 etc.

Q.6 What is bit and why binary number system is important for our computer? 09502015

Ans: Bit stands for **Binary Digit**. A bit is the smallest unit of data and has value 1 or 0 representing **ON** or **OFF** state.

Computer understands only machine language, which consists of binary codes 0 and 1. So binary number system is very important for our computer.

Q.7 What is byte or character? 09502016

Ans.: A collection of 8 bits is called a byte. It is a set of bits, which represents a particular character or symbol. In memory one byte can store only one character.

Conversion

Q.8 Convert 2 + 2 = 4 into ASCII code.

Solution: 09502017

Character	Decimal Code	Binary code
2	50	00110010
+	43	00101011
2	50	00110010
=	61	00111101
4	52	00110100

So the message in ASCII is 00110010 00101011 00110010 00111101 00110100

Q.9 Convert $(ABCD)_{16}$ to binary. 09502018

Ans.

Hexa-Decimal	Binary
A	1010
B	1011
C	1100
D	1101
$ABCD_{16} = (1010101111001101)_2$	

Q.10 Convert $(0010110010001101001)_2$ to hexadecimal. 09502019

Ans.

Binary	Hexa-decimal
1001	9
0110	6
0100	4
0110	6
0001	1

$(0010110010001101001)_2 = (16469)_{16}$

Q.11 Convert $0B9_{(16)}$ into decimal. 09502020

Ans.

$$0B9_{(16)} = 0 \times 16^2 + B \times 16^1 + 9 \times 16^0$$
$$= 0 \times 16^2 + 11 \times 16^1 + 9 \times 16^0$$
$$= 0 \times 256 + 11 \times 16 + 9 \times 1$$
$$= 0 + 176 + 9$$
$$= 185_{(10)}$$

Q.12 What is Octal number system? 09502021

Ans. The octal numbering system is a base-8 number system that uses digits from 0 to 7 Each digit Represents a power of 8. In octal, the place values from right to left are $8^0, 8^1, 8^2$, and so on. For example, the octal number 157 means:

$1\times8^2+5\times8^1+7\times8^0=64+40+7=111_{10}$

Number System

Q.13 Convert $(100000)_2$ to decimal. 09502022

Ans.

$$= 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$
$$= 32 + 0 + 0 + 0 + 0 + 0$$
$$= (32)_{10}$$

Q.14 Convert $(C921)_{16}$ to decimal. 09502023

Ans.

$$= C \times 16^3 + 9 \times 16^2 + 2 \times 16^1 + 1 \times 16^0$$
$$= 12 \times 16^3 + 9 \times 16^2 + 2 \times 16^1 + 1 \times 16^0$$
$$= 12 \times 4096 + 9 \times 256 + 2 \times 16 + 1 \times 1$$
$$= 49152 + 2304 + 32 + 1$$
$$= (51489)_{10}$$

Q.15 Define Encoding. 09502024

Ans: Encoding is the conversion of data from one format or representation to another, usually for storage, transport, or processing. To put it simply, it converts information into a format that computers, devices, or software can understand, transfer, or store.

Q.16 Differentiate between signed and unsigned integer. 09502025

Ans: The major distinction between signed and unsigned integers is their capacity to represent both positive and negative numbers. Here's a clear contrast between the two.

Signed Integer: Can represent both positive and negative numbers, including zero.

An **unsigned integer** can only represent non-negative values.

Q.17 How real values stored in computer memory? 09502026

Ans: In computers, real values (also known as floating- point numbers) are used to represent numbers that have fractions or decimal points. These values are stored using IEEE standard, where each standard is represented by:

i. Sign bit ii. Exponent iii. Mantissa

Q.18 What are the binary Arithmetic Operations? 09502027

Ans: Binary arithmetic refers to the operations of addition, subtraction. Multiplication and division performed on binary numbers. Binary numbers are the basis of all operations in digital computers. Binary arithmetic operations are similar to decimal operations but follow binary rules.

Q.19 Describe the rule of binary addition. 09502028

Ans:

- 0+0=0
- 0+1=1
- 1+0=1
- 1+1=0

Q.20 What do you know about Booth's Algorithm. 09502029

Ans: Booth's Algorithm: This algorithm is efficient for multiplying binary numbers, especially when dealing with large numbers

or numbers with many consecutive similar bits.

Q.21 How Central Processing Unit (CPU) works? 09502030

Ans. The central processing unit (CPU) of a computer performs millions of binary multiplications every second to execute complex instructions and run programs.

Q.22 Write common types of encoding schemes. 09502031

Ans. Here are some of the most common text encoding schemes used in computers:

- **ASCII (American Standard Code for Information Interchange)**
- **Extended ASCII**
- **Unicode (Universal Code)**
- **UTF-8 (Unicode Transformation Format-8)**
- **UTF-16 (Unicode Transformation Format-16)**
- **UTF-32 (Unicode Transformation Format-32)**

Q.23 How images are stored in computer memory? 09502032

Ans. Images are made up of tiny dots called pixels. Each pixel has a color, and the combination of all these pixels forms the complete picture. Computers store images using numbers to represent these colors.

Q.24 Define Pixel. 09502033

Ans. A pixel **(Picture Element)** is the smallest unit of a digital image or display that may be independently manipulated for color and brightness. Pixels are the fundamental components of all digital images and screens, including those on smartphones, computers, and televisions.

Q.25 How colors are represented in computer memory? 09502034

Ans. In a color image each pixel's color can be represented by three numbers: **Red, Green, and Blue (RGB)**. Each of these numbers typically ranges from 0 to 255. For Example, a pixel with **RGB** values (255,0,0) will be bright red.

Q.26 Define the following terms: 09502035

- **JPEG**
- **PNG**
- **GIF**

Ans. JPEG (Joint Photographic Experts Group): Common format for photos. It compresses the image to save space but might lose some quality.

PNG (Portable Network Graphics): Supports transparency and maintains high quality without losing data.

GIF (Graphics Interchange Format): Used for simple animations and images with few colors.

Q.27 How Audios are stored in computer memory? 09502036

Ans. Audio files are stored by capturing sound waves and converting them into digital data. This process involves sampling and quantization.

Q.28 Define the following terms: 09502037

- **MP3**
- **WAV**
- **AAC**

Ans. MP3: A common format that compresses audio to save space but may lose some quality.

WAV: Uncompressed format that maintains high quality.

AAC: Used by many streaming services for high- quality audio with efficient compression.

Q.29 What do you know about frame and frame rates? 09502038

Ans. The number of frames shown per second, measured in frames per second (fps). Common frame rates are 24 fps (use in movies) and 30 fps(used in TV). Higher frame rates result in smoother motion in videos.

∴ **TV** Stands for Television

Q.30 Define the following terms: 09502039

- **Hard Disk Drives (HDD)**
- **Solid State Drive (SSD)**

- **Cloud Storage**

Ans. Hard Drives (HDD): Use spinning disks to read/write data. They offer large storage capacities.

Solid State Drives (SSD): Use flash memory for faster access times and better performance.

Cloud Storage: Storage files on remote servers accessible via the internet, providing flexibility and backup.

Topic Wise Multiple Choice Questions (Additional)

Choose the correction option.

Number Systems and Its Conversion

1. **11_{10} is an example of _____number system?** 09502040
   (a) Binary (b) Decimal
   (c) Hexadecimal (d) Octal
2. **Base of Octal Number system is?** 09502041
   (a) 2 (b) 8
   (c) 10 (d) 16
3. **MAC stands for?** 09502042
   (a) Media Access Control
   (b) Media Access Central
   (c) Media Access Configure
   (d) Media Access Connect
4. **BIT stands for?** 09502043
   (a) Binary Digit
   (b) Binary Integer
   (c) Binary Terms
   (d) Binary Value
5. **Single Precision use____bits?** 09502044
   (a) 30 (b) 31
   (c) 32 (d) None of These
6. **Double Precision use____bits?** 09502045
   (a) 64 (b) 62
   (c) 32 (d) None of These
7. **Binary Arithmetic operations use____?** 09502046
   (a) Addition (b) Subtraction
   (c) Multiplication
   (d) All of these
8. **In Binary 0s represents____?** 09502047
   (a) ON (b) OF
   (c) OFF (d) None of These
9. **In Binary 1s represents____?** 09502048
   (a) ON (b) OF
   (c) OFF (d) None of These
10. **CPU stands for?** 09502049
    (a) Central Processing Unit
    (b) Central Processing Unity
    (c) Central Processing United
    (d) None of These
11. **Brain of Computer is____?** 09502050
    (a) CPU (b) UDP
    (c) RAM (d) All of these
12. **ASCII code for 'P' is_____?** 09502051
    (a) 80 (b) 81
    (c) 82 (d) All of these
13. **Extended ASCII includes_____ characters?** 09502052
    (a) 64 (b) 128
    (c) 256 (d) None of these
14. **UTF stands for?** 09502053
    (a) Unicode Transformation Formula
    (b) Unicode Transformation Formation
    (c) Unicode Transformation Format
    (d) None of These
15. **Pixel stands for?** 09502054
    (a) Picture Element
    (b) Picture Identity
    (c) Picture Digit
    (d) None of These
16. **MP3 is a (an) ____file format?** 09502055
    (a) Video (b) Audio
    (c) Image (d) None of These
17. **Number "17" is equal to _____ in binary system.** 09502056
    (a) 10000 (b) 10110
    (c) 10001 (d) 10100
18. **1 Petabyte is equal to _____.** 09502057
    (a) $(1,024)^4$ bytes (b) $(1,024)^6$ bytes
    (c) $(1,024)^5$ bytes (d) $(1,024)^7$ bytes
19. **Hexadecimal system has total ___ numbers.** 09502058
    (a) 17 (b) 16
    (c) 18 (d) 15

20. **In primary and secondary storages, data is stored in the form of:** 09502059
(a) Bytes (b) Bit
(c) Nibble (d) GB

21. **Which number system consists of 0s and 1s?** 09502060
(a) Decimal (b) Hexa
(c) Binary (d) Octal

22. **Which number system has base 10 as it uses ten digits from 0 to 9?** 09502061
(a) Decimal (b) Hexa
(c) Binary (d) Octal

23. **Which number system has base 2?** 09502062
(a) Decimal (b) Hexa decimal
(c) Binary (d) Octal

24. **Which number system has base 16 ?** 09502063
(a) Decimal (b) Hexa decimal
(c) Binary (d) Octal

ASCII & Memory Units

25. **A number system is the system for representation of ____ data.** 09502064
(a) Value (b) Boolean
(c) Truth (d) Numeric

26. **The binary value of the letter 'A' is 01000001 and its decimal value is:** 09502065
(a) 67 (b) 66
(c) 69 (d) 65

27. **To convert a decimal number to binary, we divide the number by-____and take quotient and remainder.** 09502066
(a) 8 (b) 16
(c) 2 (d) 12

28. **All the characters on your keyboard has an associated code in binary. This code is called:** 09502067
(a) BCD (b) Unicode
(c) ASCII (d) EBCDIC

29. **ASCH stands for:** 09502068
(a) American Standard code
(b) American Standard Code for Information Interchange.
(c) Information code
(d) Standard institute

30. **The smallest amount of data to be stored in computer's memory is a 0 or 1 is called:** 09502069
(a) Byte (b) KB
(c) Bit (d) GB

31. **Which is a group of eight bits, enough space to store single ASCII character?** 09502070
(a) Byte (b) KB
(c) Bit (d) GB

32. **(1,024) TB or $(1,024)^5$ bytes is equal to:** 09502071
(a) 1 GB (b) 1TB
(c) 1 KB (d) 1 PB

33. **1KB = _____bytes.** 09502072
(a) 1,024 (b) 200
(c) 300 (d) 400

34. **1MB = _____bytes.** 09502073
(a) $(1,024)^5$ bytes
(b) (1,024) KB or $(1,024)^2$
(c) 3000
(d) 400

35. **1GB = (1,024) MB or _____bytes.** 09502074
(a) $(1,024)^5$ bytes
(b) (1,024) KB or $(1,024)^2$
(c) 8
(d) $(1,024)^3$

36. **1TB = (1,024) GB or ____bytes.** 09502075
(a) $(1,024)^5$ bytes
(b) (1,024) KB or $(1,024)^2$
(c) $(1,024)^4$
(d) $(1,024)^3$

Answer Key

1	b	2	b	3	a	4	a	5	c	6	a	7	d	8	c
9	a	10	a	11	a	12	a	13	c	14	c	15	a	16	b
17	c	18	c	19	b	20	a	21	c	22	a	23	c	24	b

25	b	26	d	27	c	28	c	29	b	30	c	31	a	32	d
33	a	34	b	35	d	36	c								

Solved Exercise

Choose Correct Options:

1. What does ASCII stand for? 09502076

(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? 09502077

(a) 1101102
(b) 11011
(c) 110.11
(d) 1101 A

3. How many bits are used in the standard ASCII encoding? 09502078

(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? 09502079

(a) It uses fewer bits per character
(b) It can represent characters from many different language
(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? 09502080

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

6. What is the primary difference between signed and unsigned integers? 09502081

(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 used for positive numbers

7. In the single precision, how many bits are used for the exponent? 09502082

(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? 09502083

(a) 1.4×10^{-45} to 3.4×10^{38}
(b) 1.4×10^{-38} to 3.4×10^{45}
(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? 09502084

(a) Pixels (b) Bits
(c) Bytes (d) Nodes

10. In an RGB color model, what does RGB stand for? 09502085

(a) Red, Green, Blue
(b) Red, Gray, Black
(c) Right, Green, Blue
(d) Red, Green Brown

Answer Key

1	a	2	b	3	a	4	b	5	c	6	a	7	b	8	a	9	a	10	a

Short Questions

Q.1 What is the primary purpose of the ASCII encoding scheme? 09502086

Ans. ASCII stands for American Standard Code for information Interchange. It is character encoding standard used to represent text in computers and other devices that use text. Each letter, digit, or symbol is assigned a unique number between 0 and 127.

Q.2 Explain the difference between ASCII and Unicode. 09502087

Ans. ASCII stands for American Standard Code for information Interchange. It is character encoding standard used to represent text in computers and other devices that use text.

Unicode is a character encoding standard that aims to cover all the characters used in the world's writing systems.

Q.3 How does Unicode handle characters from different languages? 09502088

Ans. UTF-8

It is a variable-length encoding scheme, which means it can use different numbers of bytes (form1 to 4) represent a character.

Example: The letters A (which is U+0041 in Unicode) is represented as 01000000 in binary (1bytes)

UTF-16

UTF-16 is another variable-length encoding scheme, but it uses 2 or 4 bytes for each character.

Example: The letter A is represented as 00000000 01000001 in binary (2 bytes).

UTF-32

UTF-32 is a fixed -length encoding scheme. Every character, no matter what, is represented using 4 bytes.

Example: The letter A is represented as 0000000 0000000 01000001 in binary (4 bytes).

Q.4 . What is the range of values for an unsigned 2-byte integer? 09502089

Ans. The range of values for an **unsigned 2-byte integer** is: **0 to 65,535**

Q.5 Explain how a negative is represented in binary. 09502090

Ans. In order to store negative values, computers use a method called two's complement. To find the two's complement of a binary number, follow these steps:

- Invert all the bits (change 0s to 1s and 1s to 0s).
- Add 1 to the Least Significant Bit (LSB).

Q.6 What is the benefit of using unsigned integers? 09502091

Ans: Using unsigned integers offers some significant advantages, especially in situations when negative values are not necessary. One of the most significant benefits is the wider positive range they provide.

Q.7 How does the number of bits affect the range of integer values? 09502092

Ans. The number of bits controls how many distinct binary combinations can be produced, hence defining the range of potential integer values. Unsigned integers range from 0 to $2^n - 1$, whereas signed integers range from -2^{n-1} to $2^{n-1}-1$.

Q.8 Why are whole numbers commonly used in computing for quantities that cannot be negative? 09502093

Ans. In computing, whole numbers are often used to show amounts that can't be negative. For example, we use whole numbers for the number of students in a school, a person's age, and grades (if there are no negative points).

Q.9 How is the range of floating-point numbers calculated for single precision? 09502094

Ans. In this standard, 4 bytes (32 bits) are used where 1 bit is for the sign, next 8 bits for the exponent and the last 23 bits for the mantissa. The range is calculated by

exponents range and precision provided by mantissa. The exponent ranges from -126 to +127. The approximate range of values: $1.4x10^{-45}$ to $3.4x10^{38}$.

Q.10 Why is it important to understand the Limitations of floating-point representation in scientific computing? 09502095

Ans: Understanding the constraints of floating-point representation in scientific computing is critical since they impact calculation accuracy, precision, and reliability. Floating-point values have limited accuracy, which can create round-off errors during arithmetic operations, resulting in minor mistakes.

Long Questions

1. **Explain how characters are encoded using Unicode. Provide examples of characters from different languages and their corresponding Unicode points.** 09502096
   Ans. Long Question No. 8
2. **Describe in detail how integers are stored in computer memory.** 09502097
   Ans. Long Question No. 3
3. **Explain the process of converting a decimal integer to its binary representation and vice versa. Include examples of both positive and negative integers.** 09502098
   Ans. Long Question No. 1
4. **Perform the following binary arithmetic operations:** 09502099

- **Multiplication of 101 by 11.**

Ans.

101 (which is 5 in decimal)
× 11 (which is 3 in decimal)

101 (101 ×1)
+ 1010 (101 × 1, shifted one place to the left)

1111 (result)

- **Division of 1100 by 10.**

Ans. We know that,
1100 = 12 in decimal
10 = 2 in decimal
So, 12/2 = 6
Which is **110** in Binary.

5. **Add the following binary numbers:** 09502100
   (a) 101
   +110

Ans. 101
+110

1011

Note: **101 (binary) + 110 (binary) = 1011 (binary)**, which equals **11 (decimal)**.

(b) **1100**
+1011

Ans.

```
 1100
+1011
-----
10111
```

Note: 1100 (binary) + 1011 (binary) = 10111 (binary), which equals **23 (decimal).**

6. Convert the following numbers to 4-digit binary and add them: 09502101

(a) 7+(-4)

Ans.

```
  0111  (7 in binary)
+ 1100  (-4 in binary)
------
 10111
```

Note: 7 + (-4) = 3 in decimal, which is **0011** in 4-bit binary.

(b) -5+3

Ans.

```
  1011  (-5 in binary)
+ 0011  (3 in binary)
------
 1110
```

Note: -5 + 3 = -2 in decimal, which is **1110** in 4-bit binary.

7. Solve the following: 09502102

(a) 1101_2-0100_2

Ans.

```
 1101
-0100
-----
 1001
```

Note: So, **1101_2 - 0100_2 = 1001_2,** which is **9** in decimal.

(b) 1010_2-0011_2

Ans.

```
 1010
-0011
-----
 0111
```

Note: So, **1010_2 - 0011_2 = 0111_2,** which is **7** in decimal.

(c) 1000_2-0110_2

Ans.

```
 1000
- 0110
------
 0010
```

Note: So, **1000_2 - 0110_2 = 0010_2,** which is **2** in decimal.

(d) 1110_2-100_2 **Note: 1110= 14 (decimal), 100=4 (decimal)**

Ans.

```
 1110
- 100
-----
 1010
```

Note: So, **1110_2 - 100_2 = 1010_2**

Solved Activities

Activity-1

09502103

1. **Marks Conversion:** Each student will take his or her marks. From 8th grade for each subject and convert them from decimal to binary .For example. , if a student scored 85 in Math, he/she will convert 85 to binary (which is 1010101).
2. **Clock Time Conversion:** Students will be given various times of the day and asked to convert them into binary. For instance, 3:45 PM would be converted as follows:

   Hours (15)=1111

   Minutes (45)=11101
3. Write your sleeping time in binary.

Ans.

- **Marks Conversion**: 85 (decimal) = **1010101 (binary)**
- **Clock Time Conversion**: 3:45 PM = **1111:101101** (binary)
- **Sleeping Time**: Example (10:30 PM) = **10110:11110** (binary)

Activity-2

09502104

1. Work in pairs to velvet the following decimal, number to octal:45,128,64.
2. Convert these octal, numbers to decimal:57,124,301.
3. Share your answers with the class and discuss any differences.

Ans.

(a) 45 (decimal to octal)

1. Divide 45 by 8:
   45÷8=5 remainder = **5**
2. Divide 5 by 8:
   5÷8= 0 remainder = **5**

Write the remainders from bottom to top:

45 (decimal) = 55 (octal)

(b) 128 (decimal to octal)

1. Divide 128 by 8:
   128÷8=16 remainder = **0**
2. Divide 16 by 8:
   16÷8=2 remainder = **0**
3. Divide 2 by 8:
   2÷8= 0 remainder = **2**

Write the remainders from bottom to top:

128 (decimal) = 200 (octal)

(c) 64 (decimal to octal)

1. Divide 64 by 8:
   64÷8=8 remainder = **0**
2. Divide 8 by 8:
   8÷8=1 remainder = **0**

3. Divide 1 by 8:
   $1 \div 8 = 0$ remainder = **1**

Write the remainders from bottom to top:

64 (decimal) = 100 (octal)

2. Convert the following octal numbers to decimal

(a) 57 (octal to decimal)

1. Expand using powers of 8:
   $(5 \times 8^1) + (7 \times 8^0)$
   $(5 \times 8) + (7 \times 1)$
   $40 + 7 = 47$

57 (octal) = 47 (decimal)

(b) 124 (octal to decimal)

1. Expand using powers of 8:
   $(1 \times 8^2) + (2 \times 8^1) + (4 \times 8^0)$
   $(1 \times 64) + (2 \times 8) + (4 \times 1)$
   $64 + 16 + 4 = 84$
   124 (octal) = 84 (decimal)

As $8^\circ = 1$

(c) 301 (octal to decimal)

1. Expand using powers of 8:
   $(3 \times 8^2) + (0 \times 8^1) + (1 \times 8^0)$
   $(3 \times 64) + (0 \times 8) + (1 \times 1)$
   $192 + 0 + 1 = 193$
   301 (octal) = 193 (decimal)

3. Share your answers with the class and discuss any differences

If you or your classmates have different answers, check for mistakes in the division

Activity-3

09502105

Find the following values and express them in hexadecimal. Discuss your findings with your classmates:

- Minimum Age to Cast Vote
- Length of the Indus River
- Total Districts in Pakistan
- Height of K2 (the second-highest mountain in the world)
- Area of Pakistan

Ans. Perform the conversions in class room or consult your teacher.

Hint: follow L/Q # 2.

Activity-4

09502106

1. Write down the binary representation of the following decimal numbers: 2.5,7.25, and 10.5
2. Then convert these binary representations to the format single precision format.
3. Discuss with your classmates how the precision of the representation changes with the size of the number

Ans. Perform the conversions in class room or consult your teacher

Activity 5:

09502107

Practicing Binary Division

Objective: to practice and understand binary division through hands -on examples.

Instructions:

1. Form group of three to four students.
2. Each group will solve the following binary division problems:

(a) $10101_2 \div 10_2$

(b) $11100_2 \div 11_2$

(c) $100110_2 \div 101_2$

3. Write down each step of your division process clearly.
4. Present your solutions to the class, explaining each step and the reasoning behind it.

Ans. Perform the conversions in class room or consult your teacher.
Hint: follow L/Q # 6.

Activity-6

09502108

1. Write down your name.
2. Find the ASCII code for each letter in your name. You can use an ASCII table to help you.
3. Convert each ASCII code to binary.
4. Write down your name in binary!

Ans. Perform the conversions in class room or consult your teacher.
Hint: Follow S/Q # 1(Additional)

Activity: 7

09502108

Create a Pixel Art

1. Use graph paper to draw a simple image, like a smiley face.
2. Color each square (pixel) and write down the RGB values for each color used.
3. Share your pixel art and RGB values with the class.

Ans. Perform the activity in class room or consult your teacher.