1. **Explain what is meant by Binary number system.**

   The binary number system is a system of numerical notation that uses only two digits: 0 and 1 which represents off and on respectively. This system is also known as the base-2 number system because it uses a base of 2 to represent numbers.

   Each digit in a binary number represents a power of 2, starting from the rightmost digit. For example, the binary number 1010 represents the decimal value 10 because: 1 x 2^3 (8) + 0 x 2^2 (0) + 1 x 2^1 (2) + 0 x 2^0 (0) = 8 + 2 = 10

2. **Explain the importance of Binary number system in computing.**

   Most of all the machine language understands only binary. The binary number system is of critical importance in computing because it is the basis for representing all digital data in computers. Every piece of digital data, such as text, images, and videos, is ultimately represented as a sequence of 0s and 1s in binary form.

3. **Explain what is meant by Most Significant Bit (MSB) and Least Significant Bit (LSB) in a binary number.**

   In a binary number, the Most Significant Bit (MSB) is the digit that represents the highest power of 2 in the number, while the Least Significant Bit (LSB) is the digit that represents the lowest power of 2.

   For example, in the binary number 10110, the leftmost digit (1) is the MSB and represents 2^4, while the rightmost digit (0) is the LSB and represents 2^0.

   The MSB and LSB are important because they determine the value and significance of the binary number. The MSB represents the sign of a binary number in a signed number system, with 0 indicating a positive number and 1 indicating a negative number. The LSB, on the other hand, determines the granularity of the number, with a change in the LSB resulting in the smallest possible change in the value of the number.

4. **Explain what is meant by Octal number system.**

   The octal number system is a positional number system that uses a base of 8. It is also known as the base-8 number system. In octal, each digit position represents a power of 8, starting with the rightmost digit.

   The octal number system uses the digits 0 through 7 to represent numbers, similar to how the decimal system uses the digits 0 through 9. However, in octal, the place values correspond to powers of 8 instead of powers of 10.

   For example, the octal number 235 represents the decimal value:

   2 x 8^2 (128) + 3 x 8^1 (24) + 5 x 8^0 (5) = 128 + 24 + 5 = 157

   The octal Number system is widely used in computer application sectors and also in the aviation sector to use the number in the form of code. Based on octal number system applications, several computing systems are developed.

5. **Explain what is meant by Hexadecimal number system.**

Hexadecimal representation is a numbering system that uses a base of 16, and is often used in computing to represent binary numbers in a more compact and human-readable format.

In hexadecimal, the digits used are 0-9 and A-F, where A-F represent values 10-15, respectively. Each digit position in a hexadecimal number represents a power of 16, starting with the rightmost digit. For example, the hexadecimal number 3F2 represents the decimal value:

3 x 16^2 (768) + 15 x 16^1 (240) + 2 x 16^0 (2) = 768 + 240 + 2 = 1010

6. **Briefly explain the usage of Hexadecimal number system in computing.**

Commonly used in Computer programming and Microprocessors. It is also helpful to describe colors on web pages. Each of the three primary colors (i.e., red, green and blue) is represented by two hexadecimal digits to create 255 possible values, thus resulting in more than 16 million possible colors.

7. **Explain how characters are stored inside the computer in your own words.**

In computers, characters are typically stored as binary numbers, which are combinations of 0s and 1s that can be represented using electronic signals. Each character is assigned a unique binary code, known as a character encoding, which maps it to a specific binary sequence.

There are many different character encodings used in computing, but the most commonly used encoding today is Unicode. Unicode is a standard character encoding that can represent almost all of the world's written languages and scripts, and it assigns a unique code point to each character.

When a user inputs text into a computer, the characters are converted to their corresponding binary codes using the character encoding. The binary codes are then stored in the computer's memory, typically in bytes (groups of 8 bits). For example, the letter "A" in Unicode is represented by the binary code 01000001, which can be stored in a single byte of memory.

When a program needs to display the characters on the screen or print them on paper, the binary codes are converted back into their corresponding characters using the same character encoding. This process is known as character decoding.

In summary, characters are stored inside a computer as binary numbers using a specific character encoding, and are typically stored in bytes of memory. This allows computers to represent and process text data in a digital format.

8. **Explain the importance of number representation in computing.**

Binary, octal, decimal, and hexadecimal are all number systems commonly used in computing. Each system has its own strengths and weaknesses, and choosing the appropriate system can depend on the application.

Binary: Binary is the most basic number system in computing, and all data is ultimately represented in binary form inside computers. Binary is used to represent digital data and is essential for computer programming and digital circuit design.

Octal: Octal is a base-8 number system, and it is often used in computer programming to represent sets of three binary digits. Octal numbers are shorter than binary numbers, and they can be more readable and easier to work with for some applications.

Decimal: Decimal is a base-10 number system that is familiar to most people. Decimal numbers are used to represent non-integer values and are essential for arithmetic operations and data analysis.

Hexadecimal: Hexadecimal is a base-16 number system that is often used in computer programming and data analysis. Hexadecimal numbers are shorter than binary numbers and are easier to read and work with than binary or octal numbers. Hexadecimal is also useful for representing binary data in a more compact form.

Understanding these different number systems is important in computing because it allows developers to choose the appropriate representation for the data they are working with. For example, binary may be used for low-level operations, while decimal may be used for data analysis. Additionally, the ability to convert between these different number systems is essential for communicating and interoperating between different computing systems.

9. Explain what is meant by BCD representation.

BCD, or Binary Coded Decimal, is a way of representing decimal digits using binary numbers. In BCD representation, each decimal digit is represented by a 4-bit binary number.

For example, the decimal number 1234 would be represented in BCD as follows:

1 2 3 4 0001 0010 0011 0100

Notice that each decimal digit is represented using a unique 4-bit binary code. This means that BCD is a self-contained system for representing decimal numbers, without requiring the use of a separate binary-to-decimal conversion algorithm.

10. Explain Zone decimal representation.

Zone decimal representation is a number representation scheme that was used in some early computers to represent decimal numbers. In this scheme, each decimal digit is represented by a 6-bit binary code, with an additional zone bit indicating the sign of the number.

The reason why zone decimal representation is considered a failure is that it suffered from several drawbacks, which ultimately led to its replacement by other number representation schemes. Some of the drawbacks of zone decimal representation include:

- **Limited range**: Zone decimal representation could only represent numbers within a limited range, typically from -9,999,999 to +9,999,999. This limited range made it unsuitable for many applications, especially those involving large numbers.
- **Wasted space:** Zone decimal representation used 6 bits to represent each decimal digit, which was wasteful in terms of storage space. This meant that it was not an efficient way to store large amounts of data.
- **Complex arithmetic operations:** Performing arithmetic operations on numbers represented in zone decimal format was complex and time-consuming, due to the need to convert between different formats and deal with overflow and underflow issues.
- **Lack of standardization:** There was no standardization in the implementation of zone decimal representation, which led to compatibility issues between different computer systems and made it difficult to exchange data between them.

Overall, the limitations and drawbacks of zone decimal representation made it a failure as a number representation scheme.

11. Explain packed decimal representation.

Packed decimal representation, also known as packed BCD (binary-coded decimal), is a way of storing and encoding decimal numbers in a computer system using binary code.

In packed decimal representation, each decimal digit is represented by a 4-bit binary code (BCD), and multiple decimal digits are combined into groups of 2 or 4 digits. These groups are then packed into a single byte or word of data, with the high-order bits used to indicate the sign of the number.

For example, the decimal number +1234 would be represented in packed decimal as the byte sequence 0x12 0x34, where the high-order bit of the first byte is set to 0 to indicate a positive number.

While packed decimal representation has several advantages for certain types of applications, there are also some drawbacks to this encoding method:

- **Space inefficiency** - Packed decimal requires more storage space than other binary encoding methods for the same number of digits, since each decimal digit is represented by 4 bits instead of the 1-2 bits required for other binary encoding schemes.
- **Limited range of values** - Packed decimal is typically used for fixed-point arithmetic with a fixed number of decimal places. It is not suitable for representing very large or very small numbers, or for performing complex mathematical operations that require floating-point arithmetic.
- **Performance overhead** - Converting packed decimal values to and from other numeric formats can be computationally expensive and time-consuming, particularly in systems that do not have built-in support for packed decimal operations.
- **Compatibility issues** - Packed decimal is a legacy encoding scheme that is not widely used in modern computer systems. Applications that rely on packed decimal data may encounter compatibility issues when integrating with newer systems that use different encoding schemes.
- **Complexity** - Packed decimal encoding and decoding can be more complex than other encoding schemes, requiring specialized algorithms and programming techniques. This can increase development and maintenance costs for applications that use packed decimal data.

12. Explain one's compliment.

An 8-bit one's complement system uses 8 binary digits (bits) to represent numbers. In this system, the range of representable numbers is from -127 to +127. The positive numbers in this system are represented in the same way as in a regular binary system, where the leftmost bit is the most significant bit and the rightmost bit is the least significant bit.

For example, the number 5 is represented as 00000101 in an 8-bit one's complement system.

To represent negative numbers in this system, the corresponding positive number is first converted to its one's complement form by inverting (i.e., flipping) all the bits. Then, 1 is added to the result to get the final two's complement representation of the negative number.

For example, the negative number -5 in an 8-bit one's complement system is represented as 11111010, which is obtained by taking the one's complement of the positive number 00000101 and adding 1.

One potential issue with one's complement is that there are two representations for zero: positive zero (represented as all 0's) and negative zero (represented as all 1's). This can cause problems in some arithmetic operations, such as subtraction, where the result may not be unique. To avoid this issue, two's complement is often used instead, which has only one representation of zero.

13. Explain Two's compliment.

An 8-bit two's complement system uses 8 binary digits (bits) to represent numbers. In this system, the range of representable numbers is from -128 to +127. The most significant bit (leftmost bit) is used to indicate the sign of the number, where 0 represents a positive number and 1 represents a negative number.

Positive numbers in this system are represented in the same way as in a regular binary system, where the leftmost bit is the most significant bit and the rightmost bit is the least significant bit.

For example, the number 5 is represented as 00000101 in an 8-bit two's complement system.

To represent negative numbers in this system, the corresponding positive number is first converted to its one's complement form by inverting (i.e., flipping) all the bits. Then, 1 is added to the result to get the final two's complement representation of the negative number.

For example, the negative number -5 in an 8-bit two's complement system is represented as 11111011, which is obtained by taking the one's complement of the positive number 00000101 and adding 1.

Two's complement has some advantages over other methods of representing negative numbers, such as one's complement. For example, it avoids the issue of having two representations for zero, which can cause problems in some arithmetic operations. Two's complement also makes it easy to perform arithmetic operations on binary numbers, as the rules for addition and subtraction are the same for both positive and negative numbers.

14. What are the character encoding schemes

Character encoding schemes are methods used to represent characters in digital form. Some of the commonly used character encoding schemes are:

- **ASCII (American Standard Code for Information Interchange)**: It is a 7-bit encoding scheme that represents the standard characters used in the English language. ASCII assigns a unique code to each character, including letters, digits, punctuation marks, and control characters.
- **EBCDIC (Extended Binary Coded Decimal Interchange Code):** It is an 8-bit character encoding scheme used primarily in IBM mainframe systems. EBCDIC assigns a unique code to each character, including letters, digits, punctuation marks, and control characters.
- **Unicode:** It is a widely used character encoding scheme that can represent characters from almost all the languages and scripts used in the world. Unicode uses a variable-length encoding scheme, which can represent each character using one or more bytes.
- **ISO/IEC 8859:** It is a series of character encoding schemes used to represent the characters used in various languages. The ISO/IEC 8859 series includes several 8-bit encoding schemes, each designed for a specific language or group of languages.
- **UTF-8:** It is a variable-length character encoding scheme that is a part of the Unicode standard. UTF-8 can represent all the characters in the Unicode character set and is widely used in web pages and other digital content.