1. Computación: Es el proceso de usar computadoras para procesar información mediante algoritmos y programas.

Codificación: Es el proceso de convertir información o datos en un formato específico para su almacenamiento, transmisión o procesamiento. La codificación puede involucrar la representación de datos en forma de bits, caracteres u otros símbolos.

Información: Es el resultado del procesamiento de datos que proporciona significado o contexto. La información es el resultado final de la computación y la codificación.

Dato: Es una representación simbólica de un hecho, concepto o instrucción. Los datos son la materia prima que se procesa para generar información.

Alfabeto: Es un conjunto finito de símbolos, generalmente letras, utilizados para representar sonidos de un idioma en particular. En el contexto de la computación, un alfabeto puede referirse al conjunto de caracteres utilizados en la codificación de datos.

Letra: Es un símbolo gráfico que representa un sonido en un idioma escrito. En computación, las letras son parte de los alfabetos utilizados para representar texto.

Símbolo: Es un elemento que representa o significa algo más allá de su forma literal. En el contexto de la codificación y la computación, los símbolos pueden incluir letras, números, caracteres especiales y otros elementos utilizados para representar datos.

Número: Es un concepto matemático que representa una cantidad. En el contexto de la computación, los números se utilizan para representar datos numéricos y realizar cálculos.

Sistema: Se refiere a un conjunto organizado de componentes interrelacionados que trabajan juntos para lograr un objetivo común. En el contexto de la computación, un sistema puede ser hardware o software diseñado para realizar tareas específicas, como un sistema operativo o un sistema de gestión de bases de datos.

Usuario: Es una persona o entidad que interactúa con un sistema de computadora para realizar tareas específicas. Los usuarios utilizan sistemas informáticos para acceder a la información, procesar datos y realizar funciones específicas según sus necesidades y roles.

1.B) Red de Computadoras: Se refiere a un conjunto de dispositivos de computación interconectados que pueden comunicarse entre sí y compartir recursos, como archivos, impresoras y conexiones a Internet.

1. Codificación en el contexto de la informática se refiere al proceso de transformar datos para su almacenamiento y transmisión. Es una técnica fundamental que convierte información en un formato digital comprensible para los sistemas informáticos. Veamos cómo se relaciona con un ejemplo cotidiano:

Imagina que deseas enviar un mensaje a un amigo. Para que tu amigo pueda entenderlo, debes codificar tus ideas en símbolos que sean fácilmente comprensibles. En este caso, el emisor (tú) convierte sus pensamientos en signos (por ejemplo, palabras escritas) que

el receptor (tu amigo) pueda interpretar. Este proceso de codificación es esencial en la comunicación humana.

3)

a. Dispositivos físicos biestables: Son dispositivos electrónicos que tienen dos estados estables y pueden mantenerse en cualquiera de ellos durante un período de tiempo indefinido. Los flip-flops son un ejemplo común de un dispositivo biestable. Se utilizan en electrónica digital y en sistemas de computación para almacenar información binaria (0s y 1s).

b. ASCII / EBCDIC:

ASCII (American Standard Code for Information Interchange): Es un estándar de codificación de caracteres que utiliza números del 0 al 127 para representar letras, números, signos de puntuación y otros caracteres. Se utiliza ampliamente en computadoras y dispositivos electrónicos.

EBCDIC (Extended Binary Coded Decimal Interchange Code): Es otro estándar de codificación de caracteres, desarrollado por IBM. A diferencia de ASCII, EBCDIC utiliza números del 0 al 255 para representar caracteres. Se utiliza principalmente en sistemas mainframe de IBM.

c. BIT / Byte:

Bit: Es la unidad básica de información en la computación y la digitalización. Un bit puede tener uno de dos valores, generalmente 0 o 1.

Byte: Es una unidad de información digital que consta de 8 bits. Un byte puede representar 256 valores diferentes (desde 0 hasta 255 en decimal).

d. Sistema de numeración (binario / Octal/ Hexadecimal/Decimal):

Binario: Es un sistema de numeración de base 2 que utiliza solo dos dígitos, 0 y 1.

Octal: Es un sistema de numeración de base 8 que utiliza los dígitos del 0 al 7.

Hexadecimal: Es un sistema de numeración de base 16 que utiliza los dígitos del 0 al 9 y las letras de la A a la F para representar los números del 10 al 15.

Decimal: Es un sistema de numeración de base 10 que utiliza los dígitos del 0 al 9. Es el sistema de numeración que usamos en la vida cotidiana.

4)

La redundancia en la información se introduce con el objetivo principal de detectar y corregir errores que pueden ocurrir durante la transmisión o el almacenamiento de datos. Los errores pueden ser causados por una variedad de factores, como interferencias electromagnéticas, fallos de hardware, etc. Al agregar datos redundantes (es decir, datos adicionales que no aportan nueva información, pero que replican la información existente), es posible detectar cuando los datos han cambiado y, en algunos casos, corregir los errores.

El bit de paridad es un ejemplo común de cómo se utiliza la redundancia para detectar errores.

Paridad par: Cuando los datos se van a transmitir o almacenar, se cuenta el número de bits ‘1’ en los datos. Si el número de bits ‘1’ es impar, se agrega un bit de paridad ‘1’ para hacer que el número total de bits ‘1’ sea par. Si el número de bits ‘1’ ya es par, se agrega un bit de paridad ‘0’. De esta manera, el número total de bits ‘1’ siempre será par.

Paridad impar: Funciona de manera similar a la paridad par, pero en este caso se agrega un bit de paridad para asegurar que el número total de bits ‘1’ sea impar.

Cuando los datos se reciben o se recuperan del almacenamiento, se vuelve a contar el número de bits ‘1’, incluyendo el bit de paridad. Si los datos no han cambiado, el número de bits ‘1’ debería ser par (para paridad par) o impar (para paridad impar). Si no es así, entonces sabemos que ha ocurrido un error.

5) Un sistema numérico es posicional cuando el valor de cada dígito en un número depende de su posición dentro del número. En otras palabras, el valor de un dígito es determinado por el dígito en sí y por su posición en la secuencia de dígitos.

6) **Decimal a Binario:**

Divide el número decimal por 2.

Anota el residuo.

Continúa el proceso con el cociente.

Repite los pasos hasta que el cociente sea 0.

El número binario es la secuencia de residuos, leídos de abajo hacia arriba.

**Decimal a Octal:**

Divide el número decimal por 8.

Anota el residuo.

Continúa el proceso con el cociente.

Repite los pasos hasta que el cociente sea 0.

El número octal es la secuencia de residuos, leídos de abajo hacia arriba.

**Decimal a Hexadecimal:**

Divide el número decimal por 16.

Anota el residuo. Si el residuo es 10, anota A; si es 11, anota B; si es 12, anota C; si es 13, anota D; si es 14, anota E; y si es 15, anota F.

Continúa el proceso con el cociente.

Repite los pasos hasta que el cociente sea 0.

El número hexadecimal es la secuencia de residuos (o letras correspondientes), leídos de abajo hacia arriba.

7)

**Binario a Decimal:**

Comienza por el bit más a la derecha (bit menos significativo) y avanza hacia la izquierda.

Multiplica cada bit por 2n, donde n es la posición del bit comenzando desde 0 en el bit más a la derecha.

Suma todos estos valores. El resultado es el número decimal equivalente.

**Binario a Octal:**

Agrupa los dígitos binarios de derecha a izquierda en grupos de tres.

Convierte cada grupo de tres dígitos binarios en su equivalente decimal. Estos decimales serán dígitos del 0 al 7, que son los dígitos válidos en el sistema octal.

Escribe los dígitos obtenidos en el mismo orden de los grupos. El resultado es el número octal equivalente.

**Binario a Hexadecimal:**

Agrupa los dígitos binarios de derecha a izquierda en grupos de cuatro.

Convierte cada grupo de cuatro dígitos binarios en su equivalente decimal. Estos decimales serán dígitos del 0 al 15.

Para los números del 10 al 15, usa las letras A a F respectivamente.

Escribe los dígitos y/o letras obtenidas en el mismo orden de los grupos. El resultado es el número hexadecimal equivalente.

8) LINVY

9) 1001 = 11, 9

111000 = 70, 38

101 = 5, 5

1110 = 16, E

110101 = 65, 35

10)

**Octal a Binario:**

Toma cada dígito del número octal.

Convierte cada dígito octal en su equivalente binario de tres dígitos (por ejemplo, 2 en octal sería 010 en binario, 7 en octal sería 111 en binario).

Escribe los dígitos binarios obtenidos en el mismo orden de los dígitos octales. El resultado es el número binario equivalente.

**Octal a Decimal:**

Comienza por el dígito más a la derecha (dígito menos significativo) y avanza hacia la izquierda.

Multiplica cada dígito por

8n, donde n es la posición del dígito comenzando desde 0 en el dígito más a la derecha.

Suma todos estos valores. El resultado es el número decimal equivalente.

2 = 10, 2

47 = 101111, 39

651 = 110101001,425

11)

**Hexadecimal a Binario:**

Toma cada dígito (o letra) del número hexadecimal.

Convierte cada dígito o letra hexadecimal en su equivalente binario de cuatro dígitos (por ejemplo, 2 en hexadecimal sería 0010 en binario, E en hexadecimal sería 1110 en binario).

Escribe los dígitos binarios obtenidos en el mismo orden de los dígitos hexadecimales. El resultado es el número binario equivalente.

**Hexadecimal a Decimal:**

Comienza por el dígito más a la derecha (dígito menos significativo) y avanza hacia la izquierda.

Multiplica cada dígito por 16n, donde n es la posición del dígito comenzando desde 0 en el dígito más a la derecha. Para las letras A a F, usa los números 10 a 15 respectivamente.

Suma todos estos valores. El resultado es el número decimal equivalente.

A = 1010, 10

3E = 111110, 62

374 = 884, 1101110100

CB = 11001011, 203

12) 10101 -> 1010, 1011 -> 100, 1010, 1010 -> 011, 0111, 0110, 0101

21-> 10, 11 -> 1, 10, 10 -> 3, 5, 6, 7

15) a) 1, 2, 4, 8

b) verdadero, aunque el código tiene un error gravísimo. Se usa “==” para comparaciones, no “=”.