

GUIA DE LABORATORIO ARQUITECTURA DE HARDWARE

Andrés Mauricio Machado

Miguel Angel Romero

Ingeniería de software

Arquitectura de Hardware

Universidad Manuela Beltrán

Bogotá D.C.

**PREGUNTAS ORIENTADORAS**

Convertir a binario, octal y hexadecimal cada uno de los siguientes decimales.

a. 92321010 d. 34121 c. 917

**RESPUESTAS:**

**A: Binario= 11100001011001001010 octal= 540132362 Hex = E164A**

**D: BI= 1000010101001001 oct=102511 hex= 8549**

**C: BI= 1110010101 oct=1625 hex= 395**

**PRESABERES REQUERIDOS**

**╔════════════════════════════════════════════════╗**

**║ Sistemas Numéricos: ¡Un Universo Digital! ║**

**╟────────────────────────────────────────────────────╢**

**║ Aplicaciones y Uso ║**

**╟────────────────────────────────────────────────────╢**

**║ Binario │ Octal │ Decimal │**

**╟───────────────┼──────────────────┼─────────────────╢**

**║ 10101001 │ 2525 │ 169 │**

**║ 11001100 │ 3146 │ 255 │**

**║ 00110010 │ 0623 │ 832 │**

**║ 11110000 │ 7754 │ 478 │**

**║ 01010101 │ 5477 │ 690 │**

**║ 10011010 │ 4331 │ 901 │**

**╟───────────────┼──────────────────┼─────────────────╢**

**║ Hexadecimal ║**

**╟────────────────────────────────────────────────────╢**

**║ 2A3F ║**

**║ B70D ║**

**║ F8C2 ║**

**║ 6E9A ║**

**║ D531 ║**

**║ 15B8 ║**

**╟────────────────────────────────────────────────────╢**

**║ Explicación de cada Sistema Numérico ║**

**╟────────────────────────────────────────────────────╢**

**║ \*\*Binario:\*\* Fundamento de la tecnología digital.║**

**║ Utilizado en electrónica y circuitos.║**

**║ \*\*Octal:\*\* Empleado en Unix, permisos de archivos║**

**║ y programación. Base 8. ║**

**║ \*\*Decimal:\*\* Base cotidiana para matemáticas y ║**

**║ finanzas. Símbolos del 0 al 9. ║**

**║ \*\*Hexadecimal:\*\* Importante en programación y ║**

**║ sistemas de bajo nivel. Base 16. ║**

**╟────────────────────────────────────────────────────╢**

**║ Usos y Campos de Aplicación ║**

**╟────────────────────────────────────────────────────╢**

**║ \*\*Binario:\*\* Procesadores, memoria de computadora║**

**║ y sistemas de control automático. ║**

**║ \*\*Octal:\*\* Programación de sistemas Unix, ║**

**║ electrónica y comunicaciones. ║**

**║ \*\*Decimal:\*\* Cálculos cotidianos, matemáticas y ║**

**║ contabilidad financiera. ║**

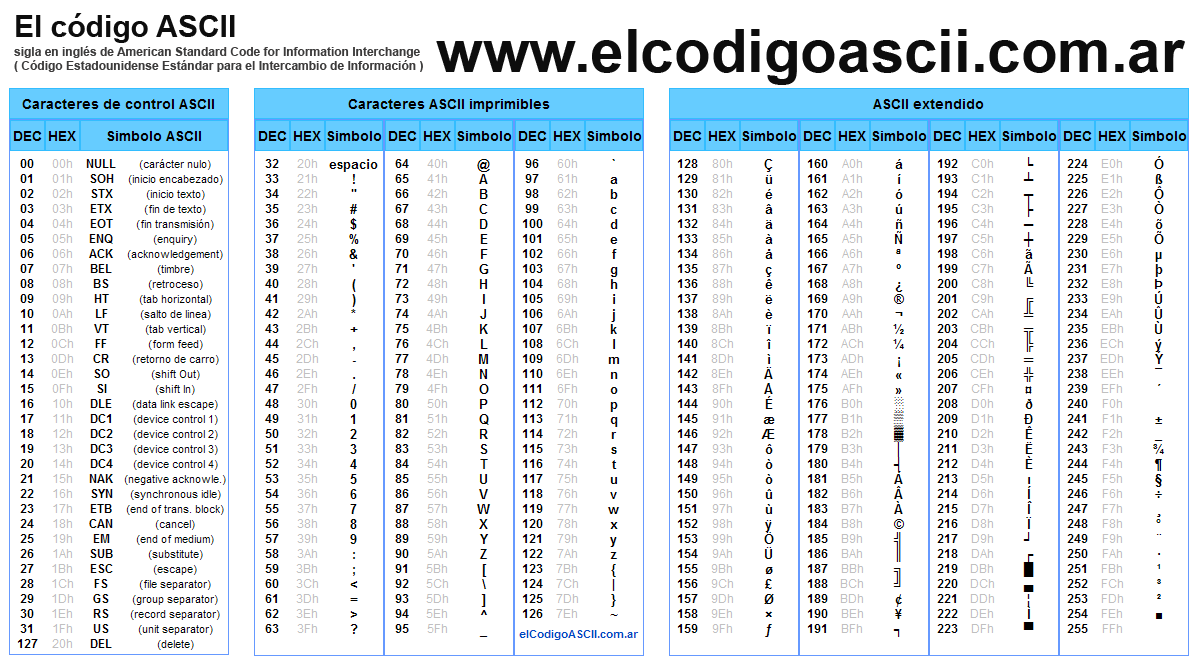
**║ \*\*Hexadecimal:\*\* Programación de bajo nivel, ║**

**║ representación de direcciones de ║**

**║ memoria y colores en diseño. ║**

**╚════════════════════════════════════════**

1. <https://youtu.be/WqhuoprnQ7k> link del video para que lo pueda ver.

2. Tabla Código ASCII[](https://elcodigoascii.com.ar/)

3.

|  |  |  |
| --- | --- | --- |
| Decimal | Hexadecimal | Binario |
| 10 | A | 1010 |
| 25 | 19 | 10011 |
| 48 | 30 | 111100 |
| 200 | C8 | 11001000 |

4.

En la aritmética de punto fijo, un número se divide en una parte entera y una parte fraccional. Por ejemplo, en un sistema de 8 bits, donde los 4 bits superiores representan la parte entera y los 4 inferiores la fraccional:

Si tomamos el número binario 11010100:

La parte entera es 1101, que en decimal es 13.

La parte fraccional es 0100, que representa 1/4 (ya que el segundo bit, contando de derecha a izquierda, corresponde a 1/4) en decimal.

Por lo tanto, 1101 0100 en nuestro sistema de punto fijo sería interpretado como 13.25 en decimal.

5.

La aritmética de punto flotante es un sistema numérico que representa números reales en computación, consiste básicamente en expresar los números en notación científica, ya que esta es útil para representar números muy grandes o pequeños. La aritmética de punto flotante lo que hace es dividir en 3 componentes dichos números expresados en notación científica.

En la aritmética de punto flotante, un número se representa mediante tres componentes:

Signo: Indica si el número es positivo o negativo.

Mantisa: También conocida como significando, es la parte fraccional del número.

Exponente: Determina la ubicación del punto decimal.

Por ejemplo, el número 123.45 se representa como 1.2345 x 10^2 en notación científica. En este caso, el signo es positivo, la mantisa es 1.2345, y el exponente es 2.

6.

6.1.

|  |  |  |  |
| --- | --- | --- | --- |
| Decimal | Binario | Octal | Hexadecimal |
| 325 | 101000101 | 505 | 145 |
| 954 | 1110111010 | 1672 | 3ba |
| 1562 | 11000011010 | 3032 | 1562 |
| 2463 | 100110011111 | 4637 | 99f |

6.2

a. 111001 = 1+8+16+32 = 57

b. 1010101 = 1+ 4+16+64 = 85

c. 11100101 = 1+4+32+64+128 = 229

d. 101011110101 = 1+4+16+32+64+128+512+2048 = 2805

6.3

a. 65 = 5 \* 8^0 + 6\* 8^1 = 53

b. 327 = 7\*8^0 + 2\*8^1 + 3\*8^2 = 215

c. 2586 = 6\*8^0 + 8\*8^1 + 5\*8^2 + 2\*8^3 = 21

d. 4050 = 0\*8^0 + 5\*8^1+ 0\*8^2 + 4\*8^3= 2088

6.4

a. 15A1 = 5537

b. 25BD1= 154577

c. CFF2 = 53234

d. 15CF2 = 89330

7. Procedimiento Sumas Binarias:

|  |  |  |
| --- | --- | --- |
| 4 Reglas Básicas | Suma | Acarreo |
| 0 + 0 = 0 | 0 | 0 |
| 0 + 1 = 1 | 1 | 0 |
| 1 + 0 = 1 | 1 | 0 |
| 1 + 1 = 10 | 0 | 1 |

1. (11111000002) + (1111102) = 10000011110
2. (010101010102) + (1112) = 1010101111
3. (100111002) + (000012) = 10011101

8. Procedimiento Restas Binarias:

|  |  |  |
| --- | --- | --- |
| 4 Reglas Básicas | Resta | Acarreo |
| 0 - 0 = 0 | 0 | 0 |
| (1) 0 - 1 = 1 | 1 | 1 |
| 1 - 0 = 1 | 1 | 0 |
| 1 - 1 = 0 | 0 | 0 |

1. (1111111) - (10101) = 1101010
2. (11100011111) - (1010110100110) = -111010000111

9. Procedimiento Multiplicación Binaria

|  |
| --- |
| 4 REGLAS BASICAS |
| 0 \* 0 = 0 |
| 0 \* 1 = 0 |
| 1 \* 0 = 0 |
| 1 \* 1 = 1 |

1. (1111011) \* (111100) = 1110011010100
2. (1111111111) \* (110) = 1011111111010

10. Las teclas que se utilizan al usar códigos ASCII tienen las siguientes funciones:

El Código Estándar Estadounidense para el Intercambio de Información, o ASCII (por sus siglas en inglés), es un método para codificar caracteres en números. En otras palabras, es una tabla en la cual los caracteres que se imprimen o se muestran corresponden a los números que los representan. Las teclas del teclado se utilizan para ingresar caracteres, y estas entradas se convierten en códigos ASCII para que el computador pueda entenderlos y codificarlos.

11 y 12.

Enlace de Interprete Python: <https://www.programiz.com/python-programming/online-compiler/>

CODIGO FUENTE:

def convertir\_base(numero, base\_original, base\_objetivo):

caracteres\_base = '0123456789ABCDEF'

numero\_decimal = 0

for indice, digito in enumerate(str(numero)[::-1]):

numero\_decimal += caracteres\_base.index(digito) \* (base\_original \*\* indice)

numero\_nueva\_base = ''

while numero\_decimal > 0:

numero\_decimal, residuo = divmod(numero\_decimal, base\_objetivo)

numero\_nueva\_base = caracteres\_base[residuo] + numero\_nueva\_base

return numero\_nueva\_base if numero\_nueva\_base else '0'

def convertir\_binario(numero, base\_original):

return convertir\_base(numero, base\_original, 2)

def convertir\_ascii(numero\_binario):

numero\_decimal = int(numero\_binario, 2)

if 32 <= numero\_decimal <= 126:

return chr(numero\_decimal)

else:

return 'El número no corresponde a un caracter ASCII imprimible.'

def principal():

numero = input("Ingrese el número: ")

base\_original = int(input("Ingrese la base original (M): "))

base\_objetivo = int(input("Ingrese la base objetivo (N): "))

if not 1 <= base\_original <= 16 or not 1 <= base\_objetivo <= 16:

print("Las bases deben estar entre 1 y 16.")

return

print(f"El número convertido de la base {base\_original} a la base {base\_objetivo} es: {convertir\_base(numero, base\_original, base\_objetivo)}")

print(f"El número convertido a binario desde la base {base\_original} es: {convertir\_binario(numero, base\_original)}")

valor\_ascii = convertir\_ascii(convertir\_binario(numero, base\_original))

print(f"El valor ASCII del número es: {valor\_ascii}")

if \_\_name\_\_ == "\_\_main\_\_":

principal()