

PREGUNTAS ORIENTADORAS

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

✓ a. 923210

- BINARIO: 11100001011001001010
- OCTAL: 3413112
- HEXADECIMAL: E164A

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

✓ b. 3412

- BINARIO: 110101010100
- OCTAL:
- HEXADECIMAL:

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

✓ c. 91710

- BINARIO:
- OCTAL:
- HEXADECIMAL:

SESIÓN 1:

1. Realiza un video no mayor a 5 minutos que explique el proceso de conversión de los sistemas de numeración binario, decimal, octal hexadecimal. Revisa el link de apoyo
2. Busca y toma una imagen de la tabla de código ASCII.

El código ASCII

sigla en inglés de American Standard Code for Information Interchange
(Código Estadounidense Estándar para el Intercambio de Información)

Tomada del sitio:

tecnologiaeinformaticaa.es.tl

Caracteres ASCII de control		
00	NULL	(carácter nulo)
01	SOH	(inicio encabezado)
02	STX	(inicio texto)
03	ETX	(fin de texto)
04	EOT	(fin transmisión)
05	ENQ	(consulta)
06	ACK	(reconocimiento)
07	BEL	(timbre)
08	BS	(retroceso)
09	HT	(tab horizontal)
10	LF	(nueva línea)
11	VT	(tab vertical)
12	FF	(nueva página)
13	CR	(retorno de carro)
14	SO	(desplaza afuera)
15	SI	(desplaza adentro)
16	DLE	(esc. vínculo datos)
17	DC1	(control disp. 1)
18	DC2	(control disp. 2)
19	DC3	(control disp. 3)
20	DC4	(control disp. 4)
21	NAK	(conf. negativa)
22	SYN	(inactividad sínc)
23	ETB	(fin bloque trans)
24	CAN	(cancelar)
25	EM	(fin del medio)
26	SUB	(sustitución)
27	ESC	(escape)
28	FS	(sep. archivos)
29	GS	(sep. grupos)
30	RS	(sep. registros)
31	US	(sep. unidades)
127	DEL	(suprimir)

Caracteres ASCII imprimibles			
32	espacio	64	@
33	!	65	A
34	"	66	B
35	#	67	C
36	\$	68	D
37	%	69	E
38	&	70	F
39	'	71	G
40	(72	H
41)	73	I
42	*	74	J
43	+	75	K
44	,	76	L
45	-	77	M
46	.	78	N
47	/	79	O
48	0	80	P
49	1	81	Q
50	2	82	R
51	3	83	S
52	4	84	T
53	5	85	U
54	6	86	V
55	7	87	W
56	8	88	X
57	9	89	Y
58	:	90	Z
59	;	91	[
60	<	92	\
61	=	93]
62	>	94	^
63	?	95	_
		96	`
		97	a
		98	b
		99	c
		100	d
		101	e
		102	f
		103	g
		104	h
		105	i
		106	j
		107	k
		108	l
		109	m
		110	n
		111	o
		112	p
		113	q
		114	r
		115	s
		116	t
		117	u
		118	v
		119	w
		120	x
		121	y
		122	z
		123	{
		124	
		125	}
		126	~

ASCII extendido (Página de código 437)							
128	Ç	160	á	192	Ł	224	Ó
129	ù	161	í	193	ł	225	ô
130	é	162	ó	194	Ṭ	226	Ô
131	â	163	ú	195	ṭ	227	Ò
132	ä	164	ñ	196	—	228	õ
133	à	165	Ñ	197	+	229	Ö
134	â	166	ª	198	ä	230	Ø
135	ç	167	º	199	Ä	231	þ
136	ê	168	¿	200	Ł	232	ƒ
137	ë	169	®	201	Œ	233	Ú
138	è	170	™	202	Œ	234	Û
139	ï	171	½	203	Ṭ	235	Ü
140	î	172	¼	204	ṭ	236	ý
141	ì	173	¡	205	=	237	Ý
142	Ā	174	«	206	≠	238	˘
143	Ă	175	»	207	#	239	ˆ
144	É	176	⋮	208	ö	240	≡
145	æ	177	⋮	209	ø	241	±
146	Æ	178	⋮	210	Ê	242	_
147	ó	179		211	Ê	243	¼
148	ö	180	Œ	212	È	244	Œ
149	ô	181	Ā	213	ı	245	\$
150	û	182	Ă	214	İ	246	÷
151	ù	183	Ā	215	İ	247	°
152	ý	184	©	216	İ	248	°
153	Ö	185	Œ	217	Œ	249	°
154	Û	186		218	Œ	250	°
155	ø	187	Œ	219	■	251	°
156	£	188	Œ	220	■	252	°
157	Ø	189	¢	221	⋮	253	°
158	×	190	¥	222	ı	254	°
159	f	191	Œ	223	■	255	nbsp

3. Consulta y realiza la tabla de hexadecimal con cuatro entradas.

Decimal	Binario	Hexadecimal	Octal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17

SESIÓN 2:

4. Consulta y explica con un ejemplo la Aritmética de punto fijo

La aritmética de punto fijo es una representación numérica de coma fija que se utiliza para representar números de base pequeña con órdenes de magnitud acotados. En esta representación, el punto decimal siempre estará en la misma posición y se utiliza una cantidad fija de dígitos después del punto decimal. El último bit se utiliza para representar el signo del número.

Por ejemplo, si queremos representar el número 3.75 en una aritmética de punto fijo con 4 bits para la parte entera y 4 bits para la parte fraccionaria, primero debemos convertir el número a binario. La parte entera es 3 y se representa como 0011 y la parte fraccionaria es 0.75 que se representa como 0.11. Entonces, el número 3.75 se representa como 0011.1100.

5. Consulta y explica con un ejemplo la Aritmética de punto flotante

La aritmética de punto flotante es una representación numérica que se utiliza para representar números reales con una precisión variable. En esta representación, el punto decimal no siempre estará en la misma posición y se utiliza una cantidad variable de dígitos para la mantisa y el

exponente. Una aritmética de punto flotante se representa como: $F = (\beta, p, m, M)$ donde β representa la base del exponente, p la precisión o mantisa, m el menor exponente posible y M el mayor exponente posible.

Por ejemplo, si queremos representar el número 123456789 en una aritmética de punto flotante con 8 bits para la mantisa y 4 bits para el exponente, primero debemos convertir el número a binario. El número 123456789 en binario es 111010110111100110100010101. La mantisa es 10111100 y el exponente es 0110. Entonces, el número 123456789 se representa como 1.0111100×2^6 .

6. Realiza el proceso de las siguientes conversiones

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

✓ a. 325

- BINARIO: 101000101

- OCTAL: 505

-HEXADECIMAL: 145

b. 954

-BINARIO: 1110111010

-OCTAL: 1672

- HEXADECIMAL: 3BA

c. 1562

-BINARIO: 11000011010

- OCTAL: 3032

- HEXADECIMAL: 61A

d. 2463

- BINARIO: 100110011111

- OCTAL: 4637

- HEXADECIMAL: 99F

Convertir a decimal los siguientes binarios.

✓ a. 111001

- DECIMAL: 57

b. 1010101

- DECIMAL: 85

c. 11100101

- DECIMAL: 229

d. 101011110101

- DECIMAL: 2805

Convertir a decimal los siguientes octales.

✓ a. 65

- DECIMAL: 53

b. 327

- DECIMAL: 215

c. 2586

- DECIMAL:

d. 4050

- DECIMAL: 2088

Convertir a decimal los siguientes hexadecimales.

✓ a. 15A

- DECIMAL: 346

b. 25BD

- DECIMAL: 9661

c. CFF2

- DECIMAL: 53234

d. 15CF2

- DECIMAL: 89330

7. Realiza el procedimiento para las siguientes sumas binarias

✓ (1111100000) + (111110)

- 10000011110

✓ (01010101010) + (111)

- 01010110001

✓ (10011100) + (00001)

- 10011101

8. Realiza el procedimiento para las siguientes restas binarias

✓ (1111111) - (10101)

- 1101010

✓ (11100011111) - (1010110100110)

- 1000101111001

9. Realiza el procedimiento para las siguientes restas binarias

✓ (1111011) * (111100)

- 1110011010100

✓ (1111111111) * (110)

- 1011111111010

10. Describe la función de las teclas que se involucran al usar el código Ascii.

SESIÓN 3:

Las teclas que se involucran al usar el código ASCII son las siguientes:

- La tecla ALT: se usa para insertar un carácter ASCII manteniendo presionada la tecla ALT mientras se escribe el código de carácter en el teclado numérico².

- La tecla X: se usa para insertar un carácter Unicode escribiendo el código de carácter, presionando ALT y luego presionando X.

- Las teclas de *función (F1-F12)*: pueden generar una secuencia corta de caracteres, a menudo comenzando con «Escape» (ASCII 27) o tecla Esc, o bien enviar una secuencia de caracteres que es interpretada por el sistema operativo o algunos programas para iniciar rutinas o tareas predeterminadas.

11. Realice un algoritmo que permita pasar un número de una base sea cual sea "M" a

otra base "N", Nota M y N tienen como valor mínimo 1 y como valor máximo 16,

además debe pasar el número a binario y entregar el mensaje ascii.

/*

* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template

*/

```

package javaapplication14;

import java.util.Scanner;

public class BaseConversion {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Ingrese el numero: ");
        String numeroStr = scanner.nextLine();

        System.out.print("Ingrese la base M (entre 1 y 16): ");
        int baseM = scanner.nextInt();

        System.out.print("Ingrese la base N (entre 1 y 16): ");
        int baseN = scanner.nextInt();

        if (baseM < 1 || baseM > 16 || baseN < 1 || baseN > 16) {
            System.out.println("Bases invalidas. Deben estar entre 1 y 16.");
            return;
        }

        if (!validarDigitos(numeroStr, baseM)) {
            System.out.println("Los digitos ingresados no son validos para la base M.");
            return;
        }

        int decimal = Integer.parseInt(numeroStr, baseM);

        String resultadoBaseN = Integer.toString(decimal, baseN);
        String binario = Integer.toBinaryString(decimal);
        char caracterAscii = (char) decimal;

        System.out.println("Numero en base " + baseN + ": " + resultadoBaseN);
        System.out.println("Representacion binaria: " + binario);
        System.out.println("Valor ASCII: " + caracterAscii);
        System.out.println("Caracter ASCII: " + caracterAscii);
    }

    public static boolean validarDigitos(String numeroStr, int base) {
        String digitosValidos = "0123456789ABCDEF";

        for (int i = 0; i < numeroStr.length(); i++) {
            char digito = Character.toUpperCase(numeroStr.charAt(i));
            if (digitosValidos.indexOf(digito) == -1 || digitosValidos.indexOf(digito) >= base) {
                return false;
            }
        }
    }
}

```

```

    }

    return true;
}
}

```

SESIÓN 4:

12. Implemente el algoritmo en el lenguaje que desee y entregue dicho programa ejecutable.

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows a project named 'PROYECTO LOGICA 2DO CORTE' with several source packages and libraries.
- Source Editor:** Displays the code for 'BaseConversion.java'. The code includes methods for converting a decimal number to a specified base and validating if the digits are within the allowed range for that base.
- Output Window:** Shows the execution results of 'JavaApplication13 (run) #2'. It displays user input and the program's output, including a validation message.

Code in BaseConversion.java:

```

28     System.out.println("Los digitos ingresados no son validos para la base M.");
29     return;
30 }
31
32 int decimal = Integer.parseInt(numeroStr, radix: baseN);
33
34 String resultadoBaseN = Integer.toString(decimal, radix: baseN);
35 String binario = Integer.toBinaryString(decimal);
36 char caracterAscii = (char) decimal;
37
38 System.out.println("Numero en base " + baseN + ": " + resultadoBaseN);
39 System.out.println("Representacion binaria: " + binario);
40 System.out.println("Valor ASCII: " + caracterAscii);
41 System.out.println("Caracter ASCII: " + caracterAscii);
42 }
43
44 public static boolean validarDigitos(String numeroStr, int base) {
45     String digitosValidos = "0123456789ABCDEF";
46
47     for (int i = 0; i < numeroStr.length(); i++) {
48         char digito = Character.toUpperCase(numeroStr.charAt(i));
49         if (digitosValidos.indexOf(digito) == -1 || digitosValidos.indexOf(digito) >= base) {
50             return false;
51         }
52     }
53 }

```

Output Window Content:

```

JavaApplication13 (run) #2
Ingreso el numero: 200
Ingreso la base M (entre 1 y 16): 2
Ingreso la base N (entre 1 y 16): 1
Los digitos ingresados no son validos para la base M.
BUILD SUCCESSFUL (total time: 6 seconds)

```