

---

# Guía de Ejercicios Prácticos Sistemas de Numeración

---

1. Pasar los siguientes números de la base indicada a la base 10.

a.  $19_{16}$       b.  $0,68_9$       c.  $3,(57)_8$       d.  $15_{15}$       e.  $0,CAFE_{16}$       f.  $7,B_{16}$

g.  $0,2(13)_4$       h.  $0,111_2$       i.  $654_8$       j.  $0,(2)_3$       k.  $53_{14}$       l.  $123_8$

Nota: Entre paréntesis se consignan los dígitos periódicos.

a.  $19_{16} = 1 \times 16^1 + 9 \times 16^0_{10} = 25_{10}$

b.  $0,76543_{10}$       c.  $3,74603_{10}$       d.  $20_{10}$       e.  $0,79293_{10}$       f.  $7,6875_{10}$

g.  $0,61(6)_{10}$       h.  $0,875_{10}$       i.  $428_{10}$       j.  $1_{10}$       k.  $73_{10}$       l.  $83_{10}$

2. Pasar los siguientes números de la base 10 a la base indicada.

a.  $19 \dots\dots]_6$       b.  $5,35\dots]_9$       c.  $6,(21)\dots\dots]_8$       d.  $5\dots\dots]_{15}$       e.  $0,56\dots\dots]_{16}$

f.  $7,48\dots\dots]_{16}$       g.  $0,(3)\dots]_3$       h.  $45\dots\dots]_8$       i.  $13\dots\dots]_{14}$       j.  $9,15(2)\dots\dots]_5$

a.  $31_6$

b.  $5,(31)_9$

$$\begin{array}{r} 19 \overline{) 6} \\ \underline{1 \phantom{0}} \phantom{0} \\ 5 \phantom{0} \end{array}$$

←

b)  $5,35_{10} \rightarrow _9 = 5,(31)_9$

$$\begin{array}{r} 0,35 \\ \times 9 \\ \hline 3,15 \end{array} \quad \begin{array}{r} 0,15 \\ \times 9 \\ \hline 1,35 \end{array} \quad \begin{array}{r} 0,35 \\ \times 9 \\ \hline 3,15 \end{array}$$

c.  $315/41_8$

d.  $5_{15}$

e.  $0,(8F5C2)_{16}$

f.  $7,(7AE14)_{16}$

g.  $0,1_3$

h.  $55_8$

i.  $D_{14}$

j.  $14,033\dots]_5$

3. Pasar los siguientes números a las bases indicadas.

a.  $45]_8 \dots]_7$    b.  $1010,01]_2 \dots]_9$    c.  $0,(1A)]_{16} \dots]_8$    d.  $54321]_6 \dots]_8$    e.  $7,B]_{16} \dots]_{11}$

f.  $0,2(3)]_7 \dots]_9$    g.  $0,111]_2 \dots]_5$    h.  $56]_8 \dots]_3$    i.  $3,(2)]_4 \dots]_2$    j.  $ABC]_{16} \dots]_5$

a.  $52]_7$    b.  $11,(2)]_9$    c.  $0,(06415032)]_8$    d.  $16451]_8$    e.  $7,(7620)]_{11}$

f.  $5/15]_9$    g.  $0,(41)]_5$    h.  $1201]_3$    i.  $11,(10)]_2$    j.  $41443]_5$

4. Pasar los siguientes números a las bases indicadas usando propiedad de potencia o raíz de base.

a.  $75]_8 \dots]_{16}$    b.  $BCD,EF]_{16} \dots]_2$    c.  $2211]_{3} \dots]_9$    d.  $1000,01]_{2} \dots]_8$    e.  $1001]_{2} \dots]_{16}$

a.  $3D]_{16}$    b.  $101111001101,11101111]_2$    c.  $84]_9$    d.  $10,2]_8$    e.  $9]_{16}$

5. Realizar las siguientes sumas en la base indicada:

a.  $101 + 1010]_2$    b.  $5423 + 2134]_8$    c.  $CDDE + 1F1F]_{16}$

d.  $1011 + 1111]_2$    e.  $1213 + 2113]_4$    f.  $7231 + 3025]_8$

a. 1111 (base 2)   b. 7557 (base 8)   c. ECFD (base 16)

d. 11010 (base 2)   e. 3332 (base 4)   f. 12256 (base 8)

6. Realizar las siguientes restas en la base indicada:

a.  $1101 - 10]_2$

b.  $5423 - 1111]_8$

c.  $DDE - F1F]_{16}$

d.  $1011 - 1111]_2$

e.  $A213 - 2F1B]_{16}$

f.  $1111 - 1011]_2$

g.  $245 - 685]_{10}$

h.  $A32F - CD4E]_{16}$

i.  $10011101 - 11001110]_2$

a.  $1011$  (base 2)

b.  $4312$  (base 8)

c.  $-141$  (base 16)

d.  $-100$  (base 2)

e.  $72F8$  (base 16)

f.  $100$  (base 2)

7. Si estas sumas son correctas en alguna base, indicar en cuál.

a.  $6 + 7 = 11$

b.  $5 + 7 = 13$

c.  $5 + 7 = 17$

8. Dados los siguientes números, almacenarlos en formato binario de punto fijo sin signo de 8 y 16 bits e indicar su configuración decimal y hexadecimal.

a.  $23]_{10}$

b.  $8320]_9$

c.  $72114]_{11}$

a)  $23]_{10}$  00010111 BPF s/s 8b  
0000000000010111 BPF s/s 16b

$$\begin{array}{r}
 23 \mid 2 \\
 \hline
 1 \ 11 \mid 2 \\
 \hline
 1 \ 5 \mid 2 \\
 \hline
 1 \ 2 \mid 2 \\
 \hline
 0 \ 1
 \end{array}$$

Ahora hallamos la configuración decimal y hexadecimal en cada formato. Para ello pasamos de base  $00010111]_2$  a base 10 y 16:

$$00010111]_2 = 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 1 \times 2^4]_{10} = 23]_{10}$$

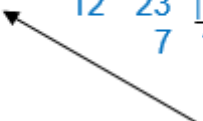
$$00010111]_2 \Rightarrow \text{Agrupamos de 4 dígitos binarios para obtener uno hexadecimal} \Rightarrow 17]_{16}$$

$$0000000000010111]_2 = 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 1 \times 2^4]_{10} = 23]_{10}$$

$0000000000010111]_2 \Rightarrow$  Agrupamos de a 4 dígitos binarios para obtener uno hexadecimal  $\Rightarrow 0017]_{16}$

b)  $8320]_9$

$$8320]_9 = 0 \times 9^0 + 2 \times 9^1 + 3 \times 9^2 + 8 \times 9^3]_{10} = 6093]_{10}$$

$$\begin{array}{r} 6093 \quad | 16 \\ D \leftarrow 13 \quad 380 \quad | 16 \\ \quad 12 \quad 23 \quad | 16 \\ \quad \quad 7 \quad 1 \end{array}$$


$17CD]_{16}$

	8bits sin signo	Conf. 10	Conf. 16	16bits sin signo	Conf. 10	Conf. 16
a.	$0001\ 0111]_2$	$23]_{10}$	$17]_{16}$	$0000\ 0000\ 0001\ 0111]_2$	$23]_{10}$	$17]_{16}$
b.	NO CABE	-	-	$0001\ 0111\ 1100\ 1101]_2$	$6093]_{10}$	$17CD]_{16}$
c.	NO CABE	-	-	NO CABE	-	-

9. Dadas las siguientes configuraciones de binarios de punto fijo sin signo de 32 bits, indicar el número que se encuentra almacenado en base 10.

a.  $C66EA940]_{16}$

b.  $276012]_8$

c.  $F64C6F5B]_{16}$

c.  $F64C6F5B]_{16}$

Paso de la configuración base 16 al formato base 2 (de base 16 a base 2).

$1111\ 0110\ 0100\ 1100\ 0110\ 1111\ 0101\ 1011]_2$  PF de 32 bits.

Pasar de PF simple precisión a decimal

**10.** a. Dados los siguientes números, almacenarlos en formato binario de punto fijo con signo de 8 y 16 bits e indicar su configuración decimal y hexadecimal.

**a.**  $23]_{10}$

**b.**  $8320]_9$

**c.**  $72114]_{11}$

b. Convertir al sistema binario con 6 bits los siguientes números que están en base 10 y operar considerando a los números convertidos como enteros con signo. Indicar los bits de estado NZCVP:

$26+19$   $26+32$   $26-19$   $26-26$   $19-26$   $-26+19$

$-26+26$   $-19+26$   $-19-26$   $-19-30$   $-19-31$   $-19-32$

a)

	BPF c/s 8b	Conf. 10	Conf. 16	BPF c/s 16b	Conf. 10	Conf. 16
a.	$0001\ 0111]_2$	$23]_{10}$	$17]_{16}$	$0000\ 0000\ 0001\ 0111]_2$	$23]_{10}$	$17]_{16}$
b.	NO CABE	-	-	$0001\ 0111\ 1100\ 1101]_2$	$6093]_{10}$	$17CD]_{16}$
c.	NO CABE	-	-	NO CABE	-	-

**11.** a. Ídem ejercicio 8 para binario de punto fijo con signo.

b. Convertir al sistema binario con 16 bits los siguientes números que están en base 16 y operar a los números convertidos como enteros con signo. Indicar los bits de estado NZCVP:

1. C66E - A940

2. A940 - 6F5B

3. F64C - 6F5B

4. C66E + A940

**12.** Dados los siguientes números, almacenarlos en binario de punto flotante IEEE 754 de precisión simple e indicar cuál es su configuración binaria y hexadecimal.

**a.**  $0,100111]_2$     **b.**  $0,0311]_{10}$     **c.**  $93,F1]_{16}$

**a.**  $0,100111]_2$

Para resolver este ejercicio lo que hay que hacer es lo siguiente:

a) El primer bit tomará el valor del signo del número. ( $0,100111 \Rightarrow + \Rightarrow 0$ )

b) Pasar a binario la mantisa decimal.

(En este caso ya está en binario)

c) Normalizar. Correr la coma a derecha o izquierda hasta convertir el número binario en un número de la forma  $1,.....$

El número de desplazamientos va a dar valor al exponente de forma que:

Desplazamiento a la derecha  $\Rightarrow$  Exponente negativo

Desplazamiento a la izquierda  $\Rightarrow$  Exponente positivo

$0,100111_2 \Rightarrow 1,00111 \Rightarrow$  Exponente = -1

-1 expresado en exceso 127  $\Rightarrow 126 \Rightarrow 01111110_2$

d) Mantisa representada con bit implícito  $\Rightarrow 1,00111 \Rightarrow 00111$  (el bit 1 de la parte entera no se representa)

e) El número final es  $0\ 01111110\ 00111000000000000000000_2$  (Se agregan a la derecha los "0" necesarios para completar los 23 bits de la mantisa)

Pasado a hexadecimal  $0\ 011\ 1111\ 0\ 001\ 1100\ 0000\ 0000\ 0000\ 0000_2 = 3F1C0000_{16}$

	Conf. 2	Conf. 16
a.	$0011\ 1111\ 0001\ 1100\ 0000\ 0000\ 0000\ 0000]_2$	$3F\ 1C\ 00\ 00]_{16}$
b.	$0011\ 1100\ 1111\ 1110\ 1100\ 0101\ 0110\ 1101]_2$	$3C\ FE\ C5\ 6D]_{16}$
c.	$0100\ 0011\ 0001\ 0011\ 1111\ 0001\ 0000\ 0000]_2$	$43\ 13\ F1\ 00]_{16}$

**13.** Dados los siguientes números, almacenarlos en binario de punto flotante IEEE 754 de precisión doble e indicar cuál es su configuración decimal y octal.

**a.**  $111001,001]_2$

**b.**  $29FE3,F]_{16}$

**c.**  $-2145,85 \times 10^7]_{10}$

	Conf. 8	Conf. 10
a.	401 144 400 000 000 000 000] <sub>8</sub>	4 633 236 446 331 797 504] <sub>10</sub>
b.	404 047 761 760 000 000 000] <sub>8</sub>	4 685 150 023 128 711 168] <sub>10</sub>
c.	1 404 121 362 000 000 000 000] <sub>8</sub>	13 909 982 213 572 657 152] <sub>10</sub>

**14.** Indique los pasos que serán necesarios para obtener la configuración binaria del siguiente número, expresado como binario de punto flotante IEEE 754 de precisión simple. Indicar que hacer y para qué.

$$9B9A,36C8 \times 11^{10}]_{14}$$

1) Para poder llegar a obtener un número normalizado en base 2 será necesario correr la coma y para hacer esto la base del exponente tiene que ser un 10.

Entonces hay que ver en que base el 11 en base 14 se transforma en un 10. Para ello pasamos 11 (base 14) a base 10  $\Rightarrow$  15(base 10) que significa que para obtener un 10 tengo que pasar todo el número a base 15. Entonces se pasa 9BD9A,36C78 (base 14) a base 15, el 11 (base 14) a base 15 y el exponente 10(base 14) a base 15.

2) Una vez que está todo en base 15 (quedará algo de la forma dddd,dd x 10<sup>exp</sup>) se corre la coma hasta que quede un número sin la parte exponencial.

(Ojo, este ejercicio normalmente se pide en forma teórica por lo que no habrá que hacer el corrimiento numéricamente sino solo explicar como se haría).

3) Una vez que tengo un número sin parte exponencial en base 15 de la forma ddd,ddd (base 15) lo que hago es pasarlo a base 2 y luego lo normalizo y lo meto en el formato de la forma habitual.

**15.** Indicar los números máximos y mínimos, positivos y negativos, que pueden ser almacenados en el formato flotante IEEE 754 de precisión simple.

Máximo positivo:  $1,111\ 1111\ 1111\ 1111\ 1111\ 1111 \times 10^{111\ 1111}]_2$

Mínimo positivo:  $1 \times 10^{-111\ 1110}]_2$

Mínimo negativo:  $-1 \times 10^{-111\ 1110}]_2$



Máximo negativo:  $-1,111\ 1111\ 1111\ 1111\ 1111\ 1111 \times 10^{111\ 1111}]_2$

**16.** Dadas las siguientes configuraciones de binarios de punto flotante IEEE 754 de precisión simple, indicar el número almacenado en base 10.

- a)  $86157840]_{16}$       b)  $321200235]_8$       c.  $00000000]_{16}$   
d.  $2147483648]_{10}$       e.  $86E4785A]_{16}$       f.  $1774000000]_8$

- Pasar de la configuración dada a la configuración en la cual se representa el formato (en este caso como el formato es IEEE 754 la configuración es la binaria).
- Identificar el signo observando el primer bit. Si es 0 positivo y si es 1 negativo.
- Con los bits del 2 al 9 (por ser precisión simple) hallar el exponente E. Pasarlo a decimal y restarle 127.
- Con los bits número 10 en adelante (últimos 23 bits por ser precisión simple) de la mantisa agregarle a la izquierda el bit implícito y la coma. Convertirlo a decimal para tener M.
- Expresar el número como  $\pm M \times 2^E$ . Ese número está en base 10. Puede ponerse en la calculadora para que quede en notación científica..

a)  **$86157840]_{16}$**

$86157840]_{16}$  a base 2  $\rightarrow$  = 10000110000101010111100001000000

Signo: – (negativo)

Mantisa normalizada: 1,00101010111100001000000

Exponente:

$00001100]_2 = 1100]_2 = 12]_{10}$

$\Rightarrow \text{Exp} = 12 - 127 = 115]_{10}$

Notación científica:  $1,00101010111100001000000]_2 \times 2^{115}]_{10}$

**b)  $321200235]_8$**

011 010 001 010 000 000 010 011 101

Signo: 0 positivo

Exponente: 11010001 = 209 en exceso 127

$209 - 127 = 82$  en base 10

Mantisa:  $1,010\ 000\ 000\ 010\ 011\ 101 = 2^0 + 2^{(-2)} + 2^{(-11)} + 2^{(-14)} + 2^{(-15)} + 2^{(-16)} + 2^{(-18)} = 1,250598907$

El número es  $+1,010\ 000\ 000\ 010\ 011\ 101]_2 \times 2^{82}_{10}$  con la mantisa en base 2 y la base y el exponente en base 10

El número decimal es  $+ 1,250598907 \times 2^{82} = 6,047525237 \times 10^{24}$

a.  $-2,81121 \times 10^{-35}]_{10}$

b.  $5,78938 \times 10^{-37}]_{10}$

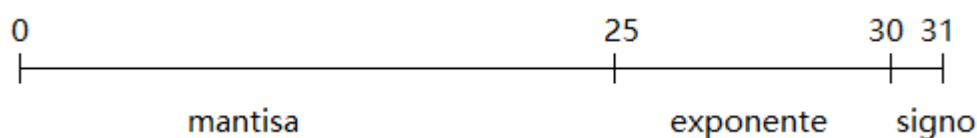
 $c. 0]_{10}$ 

d.  $-0]_{10}$

$$\text{e. } -3,69113 \times 10^{-25}]_{10}$$

### f. INFINITO POSITIVO

**17.** Expresar en base 2, los máximos y mínimos números almacenables en 32 bits de un binario de punto flotante cuyo formato es distinto al conocido ya que los primeros 26 bits representan la mantisa, los 5 restantes el exponente en exceso y el último bit es del signo.



Nota: La mantisa debe estar normalizada en binario. Tener en cuenta el “1” implícito al igual que el formato IEEE 754 tradicional.

Mantisa (26 bits)

Característica o Exponente en exceso (5 bits)

Signo (1 bit)

Exponente  $[-2^{(n-1)} ; + 2^{(n-1)} - 1] = [-16 ; +15]$

Exponente Máximo = 15 (base 10)

Exponente Mínimo = -16 (base 10)

Mínimo negativo: máxima mantisa con máximo exponente

$$-1,11111111\ 11111111\ 11111111\ 11]_2 \times 2^{15}]_{10}$$

Máximo negativo: mínima mantisa con mínimo exponente  
 $-1,0]_2 \times 2^{-16}]_{10}$

Mínimo positivo: mínima mantisa con mínimo exponente  
 $1,0]_{10} \times 2^{-16}]_{10}$

Máximo positivo: máxima mantisa con máximo exponente  
 $1,11111111 11111111 11111111 11]_2 \times 2^{15}]_{10}$

Máximo positivo:  $1,11 1111 1111 1111 1111 1111 1111 1111 \times 10^{1111}]_2$

Mínimo positivo:  $1 \times 10^{-1110}]_2$

Mínimo positivo:  $-1 \times 10^{-1110}]_2$

Máximo positivo:  $-1,11 1111 1111 1111 1111 1111 1111 1111 \times 10^{1111}]_2$

**18.** Dadas las siguientes configuraciones de empaquetados indicar que número se encuentra almacenado en base 10.

a.  $14302475]_8$

b.  $2076]_{10}$

c.  $59]_{10}$

a.  $-31 853]_{10}$

b.  $81]_{10}$

c.  $-3]_{10}$

**19.** Realizar las siguientes operaciones en la base indicada y expresar el resultado como la configuración hexadecimal y decimal de un empaquetado (si es posible).

a.  $A327 + FEC6]_{16}$

A327  
+FEC6

$1A1ED]_{16} = 106989]_{10} = 106989C]_{\text{hexadecimal empaquetado}} = 17209500]_{\text{decimal empaq.}}$

Conf. 10

a.  $17 209 500]_{10}$

b.  $890]_{10}$

Conf. 16

$1 06 98 9C]_{16}$

$03 7A]_{16}$

b.  $10210 - 3333]_4$

$0210 - -1 = 0211$

$\rightarrow -1$   
 EN 4 DIGIT 05  
 4  
 37/10

**20.** Dados los siguientes números que representan configuraciones de caracteres ASCII indicar cual es el contenido de la cadena de caracteres.

**a.** 110236461011004652523442117]<sub>8</sub>

**b.** 001010110011010100111000001110010011000000100100]<sub>2</sub>

**a.** HOLA MUNDO

**b.** +5890\$

**21.** Dados los siguientes números que representan configuraciones de caracteres EBCDIC indicar cual es el contenido de la cadena de caracteres.

**a.** 302130121000103130011232133230021131]<sub>4</sub>

**b.** D6D9C7C1F7F54BF0F3]<sub>16</sub>

**a.** IF (A>=B)

**b.** ORGA75.03

**22.** Representar los siguientes caracteres en formato EBCDIC dando su configuración octal.

**a.** 458712      **b.** G67\*fas3      **c.** ADIOS.

		EBCDIC character codes															
		1st hex digit								2nd hex digit							
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		NUL	DLE	DS		SP	&	-									0
1		SOH	DC1	SOS				/		a	j			A	J		1
2		STX	DC2	FS	SYN					b	k	s		B	K	S	2
3		ETX	TM							c	l	t		C	L	T	3
4		PF	RES	BYP	PN					d	m	u		D	M	U	4
5		HT	NL	LF	RS					e	n	v		E	N	V	5
6		LC	BS	ETB	UC					f	o	w		F	O	W	6
7		DEL	IL	ESC	EOT					g	p	x		G	P	X	7
8			CAN							h	q	y		H	Q	Y	8
9			EM							i	r	z	'	I	R	Z	9
A		SMM	CC	SM		C CENT	!		:								
B		VT	CU1	CU2	CU3		\$	,	#								
C		FF	IFS		DC4	<	*	%	@								
D		CR	IGS	ENQ	NAK	(	)	_	'								
E		SO	IRS	ACK		+	:	>	=								
F		SI	IUS	BEL	SUB		--	?	"								

a. 458712

4 5 8 7 1 2

F4 F5 F8 F7 F1 F2

F 4 F 5 F 8 F 7 F 1 F 2

1111 0100 1111 0101 1111 1000 1111 0111 1111 0001 1111 0010

111101001111010111111000111101111111000111110010

111 101 001 111 010 111 111 000 111 101 111 111 000 111 110 010

7 5 1 7 2 7 7 0 7 5 7 7 0 7 6 2

7 517 277 075 770 762]<sub>8</sub>

b. G67\*fas3

1 437 667 565 620 640 321 363]<sub>8</sub>

c. ADIOS.

6 034 231 165 561 113]<sub>8</sub>

**23.** Representar los siguientes caracteres en formato ASCII dando su configuración octal.

**a.** 458712      **b.** G67\*fas3      **c.** ADIOS.

a. 1 503 247 015 630 462]<sub>8</sub>

b. G 6 7 \* f a s 3

47 36 37 2A 66 61 73 33]<sub>16</sub>

4 7 3 6 3 7 2 A 6 6 6 1 7  
 0100 0111 0011 0110 0011 0111 0010 1010 0110 0110 0110 0001 0111

3 3 3  
 0011 0011 0011

0100011100110110001101110010101001100110011000010111001100110011  
 0 100 011 100 110 110 001 101 110 010 101 001 100 110 011 000 010 111 001 100 110  
 4 3 4 6 6 1 5 6 2 5 1 4 6 3 0 2 7 1 4 6

011  
 3

434 661 562 514 630 271 463]<sub>8</sub>  
 c. 2 024 211 123 651 456]<sub>8</sub>

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	\$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-~	63	3F	?	95	5F	~	127	7F	DEL

24. La siguiente es la configuración en base 4 de una cadena de caracteres expresada en código EBCDIC. Interpretar los caracteres de dicha cadena como la configuración en base 10 de un empaquetado e indicar el número que se encuentra almacenado en base 10.

3301331233033320331133003313

Lo que hago primero es pasar esta cadena a base 16 por propiedad de las bases obteniendo:

3 3 0 1 3 3 1 2 3 3 0 3 3 3 2 0 3 3 1 1 3 3 0 0 3 3 1 3  
 11 11 00 01 11 11 01 10 11 11 00 11 11 11 10 00 11 11 01 01 11 11 00 00 11 11 01 11  
 1111000111110110111100111111000111101011111000011110111  
 1111 0001 1111 0110 1111 0011 1111 1000 1111 0101 1111 0000 1111 0111  
 F 1 F 6 F 3 F 8 F 5 F 0 F 7  
 F1F6F3F8F5F0F7 = F1 F6 F3 F8 F5 F0 F7  
 1 6 3 8 5 0 7

		EBCDIC character codes															
		1st hex digit								2nd hex digit							
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE	DS		SP	&	-										0
1	SOH	DC1	SOS				/			a	j			A	J		1
2	STX	DC2	FS	SYN						b	k	s		B	K	S	2
3	ETX	TM								c	l	t		C	L	T	3
4	PF	RES	BYP	PN						d	m	u		D	M	U	4
5	HT	NL	LF	RS						e	n	v		E	N	V	5
6	LC	BS	ETB	UC						f	o	w		F	O	W	6
7	DEL	IL	ESC	EOT						g	p	x		G	P	X	7
8		CAN								h	q	y		H	Q	Y	8
9		EM								i	r	z		I	R	Z	9
A	SMM	CC	SM		C CENT	!		:									
B	VT	CU1	CU2	CU3		\$	,	#									
C	FF	IFS		DC4	<	*	%	@									
D	CR	IGS	ENQ	NAK	(	)	_	'									
E	SO	IRS	ACK		+	:	>	=									
F	SI	IUS	BEL	SUB		--	?	"									

Interpretando los caracteres de dicha cadena obtengo:  $1638507]_{10}$ . Ésta será la configuración en base 10 de un empaquetado.

Ahora para averiguar el número que se encuentra almacenado en base 10 paso el  $1638507]_{10}$  (config. En base 10 de un empaquetado) a base 16 y obtengo así el empaquetado  $19006B]_{16}$  (empaquetado de 3 bytes).

Por lo tanto el número almacenado en base 10 será:  $-19006]_{10}$

**25.** Dada 2013868923 configuración decimal de un número B empaquetado de 4 bytes expresar:

**a.** La configuración hexadecimal del número B representado como binario de punto fijo con signo de 32 bits.

Lo que hay que hacer es almacenar al número B (el valor en base 10 que está metido en el empaquetado) en el formato Punto Fijo c/s de 32 bits y luego dar la configuración hexadecimal de eso.

Los pasos ha realizar son los siguientes:

1.- Ver cual es el número B que tengo almacenado. Esto se hace de la siguiente manera:



2013868923 (base 10) a base 16: 7809337B (base 16) Configuración hexa del empaquetado de 4 bytes. Luego el número B es el -7809337 (base 10).

2.- Almacenar al número B en el formato punto fijo con signo de 32 bits. Esto se hace primero pasando el número a base 2 y como es negativo hallando su complemento. El resultado en base 2 sería: 11111111100010001101011011000111

3.- Pasar el resultado anterior a base 16 para obtener la configuración hexa del punto fijo de 32 bit con signo: FF88D6C7 (Configuración hexa del número B representado como Binario Punto Fijo con signo de 32 bits)

**b.** La configuración octal del número B representado como binario de punto flotante IEEE 754 de precisión simple.

1.- Ya vimos que B es -7809337 (base 10). El signo es negativo, con lo cual el primer bit del flotante será un 1.

2.- Pasamos 7809337 de base 10 a base 2 y lo normalizamos

$$11101110010100100111001 = 1,1101110010100100111001 \times 2^{22}_{10}$$

3.- Hallamos el exponente en exceso

$$\text{Exp Exceso} = \text{Exp} + 127_{10} = 22 + 127 = 149_{10} = 10010101_2$$

4.- Unimos signo, exponente en exceso y mantisa (agregamos 0 a derecha para completar precisión)

$$1\ 10010101\ 11011100101001001110010$$

5.- Pasamos a base 8

$$31273451162_8$$

**26.** Indique qué formato utilizará para almacenar el número  $2810,33_{10}$  (Justificar)

**a.** Empaquetado de 4 bytes.

**b.** Binario de punto fijo con signo de 32 bits.

**c.** Flotante IEEE 754 de precisión simple.

De los tres formatos planteados, sólo el formato Flotante IEEE 754 de precisión simple (opción c) puede almacenar números racionales no enteros (a pesar de que el  $2810,33_{10}$  no puede ser conservado en dicho formato sin pérdida de precisión).

**27.** Se tiene un nuevo formato (no IEEE 754) para almacenar binarios de punto flotante de la siguiente manera:

8 bits para el exponente, almacenado como binario de punto fijo con signo (no hay casos de exponentes especiales).

24 bits para la mantisa que debe ser almacenada normalizada en base 2, con bit 1 implícito delante de la coma.

Se pide:

- a. Determinar los máximos y mínimos números almacenables en el formato indicado.
- b. Indicar ventajas y desventajas de este formato, comparando con el formato de precisión simple IEEE 754.

Para hallarlos tenemos que saber cuáles serán los máximos y mínimos exponentes y las máximas y mínimas mantisas que maneje el formato.

Luego hay que armar la escala de representación de flotantes como se hizo para el formato tradicional.

En este formato no hay lugar para almacenar el signo del número con lo cual la escala solo tendrá la parte  $> 0$ .

El exponente se almacena directamente (sin notación en exceso). En un Binario Punto Fijo con signo de 8 bits el rango de representación es  $[-2^{(n-1)}; +2^{(n-1)}-1]$  siendo  $n$  la cantidad de bits se tiene entonces:

Mínimo Exponente:  $-2^7_{10} = -128$

Máximo Exponente:  $2^7-1_{10} = 127$

Finalmente los valores límites serán:

MÍNIMO:  $1,00..00_2 \times 2^{-128}_{10}$

MÁXIMO:  $1,11..11_2 \times 2^{127}_{10}$

a.

Máximo:  $1,1111\ 1111\ 1111\ 1111\ 1111\ 1111 \times 10^{111\ 1111}]_2$

Mínimo:  $1,0 \times 10^{-1000\ 0000}]_2$

b.

Ventaja: posee mayor precisión que el formato IEEE 754, dado que utiliza mayor cantidad de bits para exponer la mantisa.

Desventaja: no pueden expresarse números negativos en este nuevo formato.

**28.** Dada la siguiente cadena de memoria de una arquitectura IBM que se encuentran entre las direcciones  $35]_{16}$  y  $3A]_{16}$  inclusive:

330100021001022120221303

a. Indicar en qué base se encuentra.

b. Obtener el empaquetado de longitud máxima.

c. Almacenar el número hallado en **b** cómo binario de punto flotante IEEE 754 de precisión simple.

a) Para averiguar en qué base se encuentra dicha cadena realizo los siguientes pasos:

1.- Cantidad de dígitos = 24.

2.- Cantidad de celdas =  $3A]_{16} - 35]_{16} + 1]_{16} = 6]_{16} = 6]_{10}$  o bien 6 celdas.

3.- 1 celda ----- 8 bits

6 celdas ----- x = 48 bits

24 dígitos ----- 48 bits => hay 2 bits por dígito: Es base 4. La cadena se halla en base 4.

b) Obtener el empaquetado de longitud máxima.

Para obtener el empaquetado paso la cadena que se encuentra en base 4 a base 16.

Luego la cadena queda:  $F10241298A73]_{16}$

$0241298A]_{16}$  Empaquetado de 4 bytes.

c) Almacenar el número hallado en b) como binario de punto flotante de precisión simple IEEE 754.

1- Pasamos del formato empaquetado al número en base 10.

$0241298A]_{16}$  (empaquetado 4 bytes) es  $241298]_{10}$

2- Pasamos de base 10 a base 2. Pueden realizarse divisiones sucesivas por 2 o convertir primero de base 10 a base 16 y luego en forma directa a base 2.

$241298]_{10}$  es  $3AE92]_{16}$

$3AE92]_{16}$  es  $0011\ 1010\ 1110\ 1001\ 0010]_2$

3- Hay que normalizar el número.

$0011\ 1010\ 1110\ 1001\ 0010]_2 = 1,1\ 1010\ 1110\ 1001\ 0010]_2 \times 2^{17}]_{10}$

Mantisa =  $11010111010010010]_2$

4- Calculamos el exponente.

Exp.Exceso = Exponente + Exceso.

Exp.Exceso =  $17 + 127]_{10} = 144]_{10} = 10010000]_2$

5- Almacenamos el bit de signo, el exponente en exceso y los dígitos de la mantisa (agregando 0 a derecha para completar precisión):

S    Exponente                      Mantisa

0    10010000    110101110100100100000000 ]<sub>2</sub>    Binario Punto Flotante    IEEE754    Simple  
Precisión