

# Práctica 1. Sistemas numéricos de representación y Directivas.

**P1.)** Construir la tabla de transformación utilizada para convertir números binarios a hexadecimal y aplicarla en la tabla de la derecha.

Dec.	Binario N*	Hex.
0	0000	0x0
1	0001	0x1
2	0010	0x2
3	0011	0x3
4	0100	0x4
5	0101	0x5
6	0110	0x6
7	0111	0x7
8	1000	0x8
9	1001	0x9
10	1010	0xA
11	1011	0xB
12	1100	0xC
13	1101	0xD
14	1110	0xE
15	1111	0xF

Binario	Hex.	Tipo.(b,h,w)
1011	0xB	.byte
10111010	0xBA	.byte
10110011	0xB3	.byte
11111111	0xFF	.byte
110100101	0x1A5	.half
0000100100111011	0x093B	.half
0011111011000010	0x3EC2	.half
111100101000110011111011000010	0x3CA33EC2	.word
1111110110111001011101010011111	0xFEDCBA9F	.word
11111111111111111111111111111111	0xFFFFFFFF	.word

\* Usar solamente 4 bits

¿Qué representa hacer el complemento a 2 de número binario?.

Sirve para representar el negativo de un numero positivo

**P2.)** ¿Cuántos caracteres o dígitos hexadecimales tienen los tipos enteros (byte, half, word) en el procesador MIPS R2000. Y cuales son el límite máximo y mínimo que pueden representar estos tipos en decimal?.

Tipo	Número de dígitos en Hex.	S.Binario N.		C2	
		L. max	L. min	L. max	L. min
byte	2	255 ( $2^8$ )-1	0	127 ( $2^7$ )-1	-128 -( $2^7$ )
half	4	65535 ( $2^{16}$ )-1	0	32767 ( $2^{15}$ )-1	-32768 -( $2^{15}$ )
word	8	4294967295 ( $2^{32}$ )-1	0	2147483647 ( $2^{31}$ )-1	- 2147483648 -( $2^{31}$ )

**P3.)** Cuales serían las directivas más apropiadas para almacenar los siguientes datos en memoria.

Dato	Directiva
0	.byte
256	.half
33	.byte
0xEFEF	.half
0x123	.half
11010101	.byte
101010110	.half
-215	.half
0xFFFF	.half
0xABCD1234	.word
65.536e10	.float
-65.536e-50	.double
Hola	.ascii / asciiz

¿ A la hora de almacenar caracteres en memoria, qué diferencia hay entre la directiva .ascii y la directiva .asciiz, para que puede utilizarse esta diferencia?.

.ascii almacena la cadena de caracteres

.asciiz almacena la cadena de caracteres y añade un byte nulo (0x00)

El propósito de este byte nulo es identificar el fin de la cadena de texto (para que el sistema de forma automática pueda encontrar fácilmente donde finaliza la cadena)

**P4.)** Escribir el código necesario para almacenar a partir de la posición de memoria 0x10030000 del segmento de datos las siguientes constantes.

12
155
0
220.0e-11
-13
-39000
0.000000021e-10

.data 0x10030000

.byte 12, 155, 0

.float 220.0e-11

.byte -13

.word -39000

.double 0.000000021e-10

**P5.)** Partiendo del siguiente código ensamblador. Completar el valor correspondiente a cada una de las celdas de la memoria (en hexadecimal).

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

```
.data 0x10030000
.byte 15, 0xAB, 234
.asciiz "TRES"
.half 0xa123
.ascii "4"
.word 356
```

	0x10030010
0x00	0x1003000f
0x00	0x1003000e
0x01	0x1003000d
0x64	0x1003000c
0x00	0x1003000b
0x34	0x1003000a
0xa1	0x10030009
0x23	0x10030008
0x00 (.asciiz)	0x10030007
0x53 ("S")	0x10030006
0x45 ("E")	0x10030005
0x52 ("R")	0x10030004
0x54 ("T")	0x10030003
234 (0xEA)	0x10030002
0xAB	0x10030001
15 (0x0F)	0x10030000
Contenido Mem. (hex)	Dirección de Mem. (hex)

¿Qué posición de memoria ocupa el byte 0xa1 de la directiva .half 0xa123 ?  
0x10030009

¿Qué posición de memoria ocupa el carácter "R" de la Directiva .asciiz "TRES"?  
0x10030004

¿Cómo representaría el simulador la imagen de la memoria?

#### DATA

```
[0x10000000]...[0x10030000] 0x00000000
[0x10030000]          0x54eaab0f 0x00534552 0x0034a123 0x00000164
[0x10030010]...[0x10040000] 0x00000000
```

**P6.)** Usando una vez la directiva '.data', con el argumento 0x10020000, almacenar los siguientes datos en las direcciones siguientes.

Etiqueta	Dirección	dato
Var_a	0x10020000	34579
Var_b	0x10020054	0xFD
Var_c	0x10020056	0xAABB
Var_d	0x10020058	212
Var_e	0x10020059	"1FOR2"
Var_f	0x10020060	-3.6973141e28

```
.data 0x10020000
var_a: .half 34579 #(0x8713), 0x10020000
       .space 0x52
var_b: .byte 0xFD # 0x10020054
       .space 1  #(no hace falta por alineamiento del siguiente half)
var_c: .half 0xAABB # 0x10020056
var_d: .byte 212 #(0xD4) # 0x10020058
var_e: .ascii "1FOR2" # 0x10020059, (1=0x31 F=0x46 O=0x4f R=0x52 2=0x32)
var_f: .double -3.6973141e28 # 0x10020060
```