

4. Sea **0xBE333333** un número en base 16 (hexadecimal) que representa a un número real expresado en el sistema de *Punto Flotante IEEE-754 de precisión simple (32 bits)*. Indique cuál es el valor en base 10 de dicho número real.

El estándar IEEE-754 de precisión simple utiliza 32 bits para representar números reales, de los cuales:

- 1 bit se utiliza para el signo
- 8 bits se utilizan para el exponente
- 23 bits se utilizan para la mantisa

Para obtener el valor en base 10 del número 0xBE333333, primero se debe representar en binario:

B	E	3	3	3	3	3	3
1011	1110	0011	0011	0011	0011	0011	0011

Entonces, la cadena binaria resultante es 1|01111100|01100110011001100110011. Dividiendo en signo, exponente y mantisa se obtiene:

Signo (s)	Exponente (e)	Mantisa (m)
1	01111100	01100110011001100110011

El valor del **exponente** es $01111100 = 1 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5 + 1 \cdot 2^6 = 4 + 8 + 16 + 32 + 64 = 124$

Luego se aplica la fórmula:

$$n = (-1)^s * 2^{(e-127)} * (1+m)$$

donde $s=1$, $e=124$ y $m=01100110011001100110011$

Se reemplazan las variables y se obtiene:

$$n = (-1)^1 * 2^{(124-127)} * (1+01100110011001100110011)$$

$$n = -1 * 2^{(-3)} * (1,01100110011001100110011)$$

$$n = -1 * 0,001,01100110011001100110011$$

(en este paso, la coma remarcada en amarillo debe desplazarse 3 lugares hacia la izquierda, completando con ceros)

Finalmente resta pasar el número obtenido a decimal y finalmente multiplicarlo por -1 (valor que indica el signo)

0	,	0	0	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1				
0		-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16	-17	-18	-19	-20	-21	-22	-23	-24	-25	-26

Sumando únicamente aquellas potencias de 2 donde el bit es 1 se obtiene:

$$2^{-3} + 2^{-5} + 2^{-6} + 2^{-9} + 2^{-10} + 2^{-13} + 2^{-14} + 2^{-17} + 2^{-18} + 2^{-21} + 2^{-22} + 2^{-25} + 2^{-26} = 0,125 + 0,03125 + 0,015625 + (...) = 0,17499999701976776 = 0,175$$

Finalmente, el valor decimal aproximado es 0,175

5. ¿Que mensaje esta codificado en la siguiente secuencia de bytes expresados en hexadecimal: **48 6F 6C 61 20 4D 75 6E 64 6F 21**? *Respetar mayúsculas y minúsculas.*

Tabla *ASCII*:

Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex	
0	00	NUL	16	10	DLE	32	20		48	30	0	64	40	@	80	50	P	96	60	'	112	70	p
1	01	SOH	17	11	DC1	33	21	!	49	31	1	65	41	A	81	51	Q	97	61	a	113	71	q
2	02	STX	18	12	DC2	34	22	"	50	32	2	66	42	B	82	52	R	98	62	b	114	72	r
3	03	ETX	19	13	DC3	35	23	#	51	33	3	67	43	C	83	53	S	99	63	c	115	73	s
4	04	EOT	20	14	DC4	36	24	\$	52	34	4	68	44	D	84	54	T	100	64	d	116	74	t
5	05	ENQ	21	15	NAK	37	25	%	53	35	5	69	45	E	85	55	U	101	65	e	117	75	u
6	06	ACK	22	16	SYN	38	26	&	54	36	6	70	46	F	86	56	V	102	66	f	118	76	v
7	07	BEL	23	17	ETB	39	27	'	55	37	7	71	47	G	87	57	W	103	67	g	119	77	w
8	08	BS	24	18	CAN	40	28	(56	38	8	72	48	H	88	58	X	104	68	h	120	78	x
9	09	HT	25	19	EM	41	29)	57	39	9	73	49	I	89	59	Y	105	69	i	121	79	y
10	0A	LF	26	1A	SUB	42	2A	*	58	3A	:	74	4A	J	90	5A	Z	106	6A	j	122	7A	z
11	0B	VT	27	1B	ESC	43	2B	+	59	3B	;	75	4B	K	91	5B	[107	6B	k	123	7B	{
12	0C	FF	28	1C	FS	44	2C	,	60	3C	<	76	4C	L	92	5C	\	108	6C	l	124	7C	
13	0D	CR	29	1D	GS	45	2D	-	61	3D	=	77	4D	M	93	5D]	109	6D	m	125	7D	}
14	0E	SO	30	1E	RS	46	2E	.	62	3E	>	78	4E	N	94	5E	^	110	6E	n	126	7E	~
15	0F	SI	31	1F	US	47	2F	/	63	3F	?	79	4F	O	95	5F	_	111	6F	o	127	7F	DEL

Explicación de la tabla

La tabla presentada está dividida en tres columnas:

- Dec: indica el valor decimal del caracter de la tercer columna
- Hex: indica el valor hexadecimal del caracter de la tercer columna
- Caracter: indica el símbolo del caracter como X, o el nombre de aquellos caracteres especiales como NUL.

Ejemplo: la fila | 32 | 3E | > | nos indica que para el símbolo matemático > (mayor estricto) el valor decimal correspondiente es 32 y el valor hexadecimal es 3E. Claramente estos valores son equivalentes; únicamente están representados en distintas bases (10 y 16 respectivamente).

Resolución

En este ejercicio se debe utilizar la tabla ASCII para buscar los valores (en hexadecimal) del mensaje y obtener los caracteres correspondientes. Se debe respetar la distinción de mayúsculas y minúsculas de la tabla, ya que los valores son diferentes. Por ejemplo: el valor hexadecimal para 'x' es 0x78 y para 'X' es 0x58.

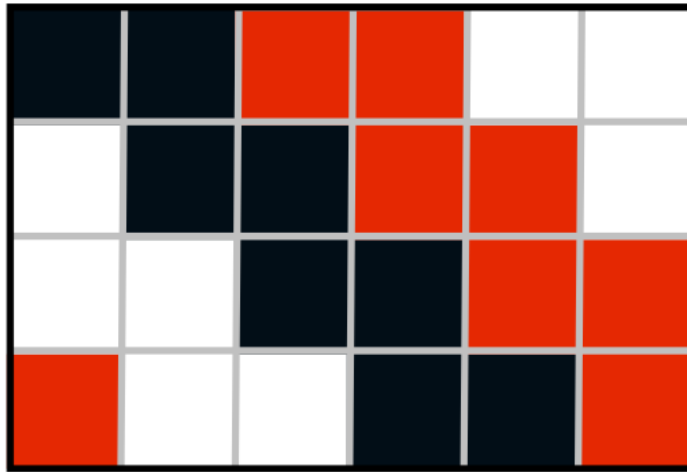
Entonces, se buscan los caracteres indicados por el mensaje (0x20 representa el espacio):

48	6F	6C	61	20	4D	75	6E	64	6F	21
H	o	l	a		M	u	n	d	o	!

Finalmente, se obtiene que el mensaje codificado es **Hola Mundo!**

6. Considerando la imagen que se muestra abajo, aplique un esquema de compresión que agrupa píxeles consecutivos de igual color y los reemplaza por una codificación “*cantidad/color*”, utilizando una codificación **3+2**, con tres bits para la cantidad y dos bits para el color. Asuma que 00 = blanco, 11 = negro y 01 = naranja.

Tenga en cuenta que al calcular la cantidad se debe considerar que las filas de la imagen son consecutivas. Es decir, si una fila termina con un grupo de píxeles un color y la siguiente comienza con píxeles del mismo color, se deben considerar todos como un solo grupo. No olvide incluir la cabecera.



Resolución

Para obtener la codificación de la imagen, debemos representar la **cabecera** y los **datos de la imagen** en binario o hexadecimal.

Para la **cabecera**, debemos utilizar 3 bytes, donde se indica: ancho, alto y profundidad de color

Ancho	Alto	Bits/píxel	Datos de la imagen...
1 Byte	1 Byte	1 Byte	

En la imagen del enunciado, tenemos:

- Ancho: 6 píxeles
- Alto: 4 píxeles
- 2 bits para indicar el color

Luego, debemos expresar dichos valores en binario, utilizando 1 byte (8 bits) para cada parte:

- Ancho: 0000 0110 en binario | 0x06 en hexadecimal | 6 en decimal
- Alto: 0000 0100 en binario | 0x04 en hexadecimal | 4 en decimal
- Prof: 0000 0010 en binario | 0x02 en hexadecimal | 2 en decimal

Finalmente, la cabecera es 0000 0110 0000 0100 0000 0010 en binario | 0x060402 en hexadecimal

Para los **datos de la imagen**, se debe codificar cada grupo de píxeles consecutivos del mismo color.

Es importante la aclaración: *“Tenga en cuenta que al calcular la cantidad se debe considerar que las filas de la imagen son consecutivas. Es decir, si una fila termina con un grupo de píxeles un color y la siguiente comienza con píxeles del mismo color, se deben considerar todos como un solo grupo”*

Luego, se obtienen los grupos de píxeles consecutivos del mismo color de la imagen:

Cant	Color
2	NEGRO
2	ROJO
3	BLANCO
2	NEGRO
2	ROJO
3	BLANCO
2	NEGRO
3	ROJO
2	BLANCO
2	NEGRO
1	ROJO

A continuación, se deben expresar dichos valores en binario o hexadecimal. Para codificar el color se debe utilizar la paleta de colores indicada (Blanco es 00, Negro es 11 y Rojo es 01). Para la cantidad deben utilizarse 3 bits como se indica en el enunciado.

		Binario	
Cantidad	Color	Cantidad	Color
2	NEGRO	010	11
2	ROJO	010	01
3	BLANCO	011	00
2	NEGRO	010	11
2	ROJO	010	01
3	BLANCO	011	00
2	NEGRO	010	11
3	ROJO	011	01
2	BLANCO	010	00
2	NEGRO	010	11
1	ROJO	001	01

Finalmente se deben indicar los **datos de la imagen** como una cadena binaria. Utilizando la información de la tabla anterior, se obtiene:

0101101001011000101101001011000101101101010000101100101

Para construir esta cadena binaria, se concatenó la información binaria de la columna Cantidad con la información de la columna Color, comenzando por la primera fila y continuando hasta el final.

Para obtener el resultado en hexadecimal, es conveniente reescribir la cadena binaria separando de a 4 bits:

0101|1010|0101|1000|1011|0100|1011|0001|0110|1101|0100|0010|1100|101

Como el último grupo tiene sólo 3 bits, debe completarse con los 0's necesarios para llegar a 4 bits. En este caso, se agrega un 0 (en amarillo, para indicar que no formaba parte de los datos originales). Finalmente se indica el valor hexadecimal correspondiente de cada grupo de 4 bits:

0101	1010	0101	1000	1011	0100	1011	0001	0110	1101	0100	0010	1100	1010
5	A	5	8	B	4	B	1	6	D	4	2	C	A

Luego, los datos de la imagen en hexadecimal son: 0x5A58B4B16D42CA

Ya se obtuvo la cabecera y los datos de la imagen. Restaría unirlos para obtener la codificación final de la imagen (en verde se marca la cabecera)

Resultado final en binario

0000011000000100000000100101101001011000101101001011000101101101010000101100101

Resultado final en hexadecimal

0x0604025A58B4B16D42CA