1ºDAW 2023/2024
11/10/2023

Representación de la Información

Sistemas Informáticos

Rodrigo Martínez Delgado

ACTIVIDADES I

- 1: ¿Cómo se clasifican los tipos de datos?

  Como datos de entrada, datos intermedios y datos de salida.
  También se distinguen entre constantes y variables cuando nos referimos a datos.

- 2: ¿Qué son los datos intermedios?

  Datos obtenidos mientras se procesan dentro del ordenador.

- 3: ¿Qué es un dato constante?

  Datos fijos cuyo valor no varía según se va procesando.

- 4: ¿Qué es un dato variable?

  Datos que varían según se van realizando las distintas operaciones o procesos dentro del ordenador.

- 5: Clasifica los siguientes tipos de datos.

  a. La edad de un alumno que se solicita por teclado.

  Dato variable.

  b. El resultado de una operación que se muestra por pantalla.

  Dato variable.

  c. G, en la ley de gravitación universal F= G M1.M2/d2.

  Dato constante.

  d. A, en la fórmula para calcular el área de un rectángulo A=base . Altura.

  Dato variable.

ACTIVIDADES II

- 6: ¿Qué es un sistema de numeración posicional?

  Aquel en cual el valor de cada símbolo viene definido por el propio símbolo y por la posición que ocupa con el resto de símbolos.

- 7: ¿Por qué decimos que el sistema decimal tiene base 10?

  Porque está basado en 10 símbolos del 0 al 9. Históricamente se debe a que el ser humano tiene diez dedos.

- 8: ¿Qué es un bit?

  Cada símbolo en el que se basa el sistema binario.

- 9: Investiga qué es un nibble.

  Conjunto de cuatro dígitos binarios.

- 10: ¿Cuántos bits son 25 MB?

  25Mb * 1.048.576 bytes/Mb = 26.214.400 bytes
  26.214.400 bytes * 8 bits/byte = **209.715.200 bits**

- 11: ¿Cuántos kilobytes tiene 1 GB?

  1Gb = 1.024Mb
  1.024Mb * 1.024Kb/Mb = **1.048.576 Kb**

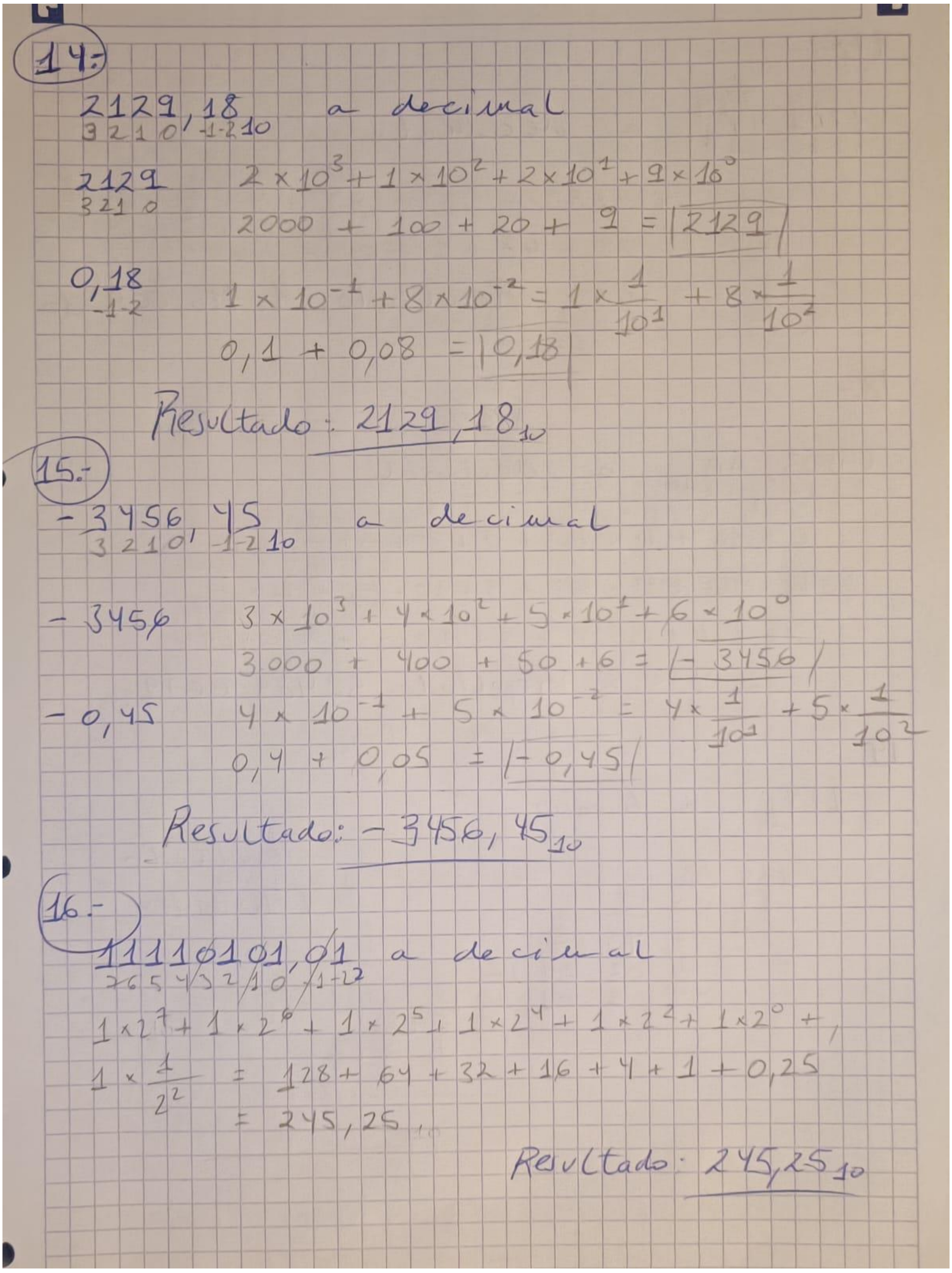
- 12: Investiga qué son los mebibytes.

  Es una unidad de información utilizada como un múltiplo del byte.
  Equivale a 2 a la 20 bytes.

ACTIVIDADES III

• 13: ¿Conoces algún ejemplo de sistema de numeración que no sea posicional?

El sistema de numeración egipcio, empleando jeroglíficos para cada orden de unidades.



14.-

$2129,18_{10}$ a decimal

3 2 1 0 , -1 -2

2129: $2\times10^3+1\times10^2+2\times10^1+9\times10^0$

$2000+100+20+9=\boxed{2129}$

$0,18$: $1\times10^{-1}+8\times10^{-2}=1\times\frac{1}{10^1}+8\times\frac{1}{10^2}$

$0,1+0,08=\boxed{0,18}$

Resultado: $2129,18_{10}$

15.-

$-3456,45_{10}$ a decimal

3 2 1 0 , -1 -2

-3456: $3\times10^3+4\times10^2+5\times10^1+6\times10^0$

$3000+400+50+6=\boxed{-3456}$

$-0,45$: $4\times10^{-1}+5\times10^{-2}=4\times\frac{1}{10^1}+5\times\frac{1}{10^2}$

$0,4+0,05=\boxed{-0,45}$

Resultado: $-3456,45_{10}$

16.-

$11110101,01$ a decimal

7 6 5 4 3 2 1 0 , -1 -2

$1\times2^7+1\times2^6+1\times2^5+1\times2^4+1\times2^2+1\times2^0+$

$1\times\frac{1}{2^2}=128+64+32+16+4+1+0,25$

$=245,25_{10}$

Resultado: $245,25_{10}$

17.-

$3261,47_8$ a decimal

3 2 1 0 -1 -2

$$3\times 8^3 + 2\times 8^2 + 6\times 8^1 + 1\times 8^0 + 4\times\frac{1}{8^1} + 7\times\frac{1}{8^2}$$

$$1536 + 128 + 48 + 1 + 0,5 + 0,1093$$

Resultado: $1.713,6093_{10}$

18.-

$CD03,AB_{16}$ a decimal

3 2 1 0 -1 -2

$$C\times 16^3 + D\times 16^2 + 0 + 3 + A\times\frac{1}{16^1} + B\cdot\frac{1}{16^2}$$

$$12\times 16^3 + 13\times 16^2 + 3 + 10\times\frac{1}{16^1} + 11\times\frac{1}{16^2}$$

$$49.152 + 3.328 + 3 + 0,625 + 0,0429$$

Resultado: $52483,6679_{10}$

19.-

174_{10} a binario

174 |2
14 87 |2
0 07 43 |2
1 03 21 |2
1 01 10 |2
1 0 5 |2
1 2 |2
0 1

Resultado

10101110_2

20.-

428_{10} a binario

428 |2								
02	214 |2							
08	014	107 |2						
0	0	07	53 |2					
		1	13	26 |2				
			1	06	13 |2			
				0	1	6 |2		
						0	3 |2	
							1	1

Resultado: 110101100_2

21.-

$001100101100,010_2$ a octal

1 4 5 4 , 2

Resultado: $1454,2_8$

22.-

$0111101110100011,0110_2$ a hexadecimal

7 11 10 3 , 6

Resultado: $7BA3,6_{16}$

23.-

$714,325_8$ a binario

5 = 101 4 = 100

2 = 010 1 = 001

3 = 011 7 = 111

Resultado:

$111001100,011010101_2$

24.-

$FA07,08_{16}$ a binario

8 = 1000 0 = 0000

0 = 0000 10 = 1010

7 = 0111 15 = 1111

Resultado:

$1111\ 1010\ 0000\ 0111,0000\ 1000_2$

25.-

$623,714_8$ a hexadecimal

6 = 110 7 = 111
2 = 010 1 = 001
3 = 011 4 = 100

Resultado: $193,E6_{16}$

$\underbrace{0001}_{1}\underbrace{1001}_{9}\underbrace{0011}_{3},\underbrace{1110}_{E}\underbrace{0110}_{6}\underbrace{0000}_{}{}_2$

26.-

$AD57,F6_{16}$ a octal

A = 1010 F = 1111
D = 1101 6 = 0110
5 = 0101
7 = 0111

Resultado: $126527,754_8$

$\underbrace{001}_{1}\underbrace{010}_{2}\underbrace{110}_{6}\underbrace{101}_{5}\underbrace{010}_{2}\underbrace{111}_{7},\underbrace{111}_{7}\underbrace{101}_{5}\underbrace{100}_{4}{}_2$

27.-

Suma binaria de 22_{10} y 14_4

$$\begin{array}{r} {}^{1\,1\;\;1} \\ 10110 \\ +\;\;1110 \\ \hline 100100_2 \end{array}$$

Resultado: 100100_2

28.-

Resta binaria de 17_{10} y 10_{10}.

$$\begin{array}{r} {}^{-1\,-1\,-1} \\ 10001 \\ -\;\;1010 \\ \hline 00111_2 \end{array}$$

Resultado: 111_2

29.-

Multiplicación binaria 29_{10} y 5_{10}

```
    11101
  ×   101
 --------
    11101
+  00000
  11101
 --------
 10010001₂
```

Resultado:

10010001_2

30.-

División binaria 42_{10} y 6_{10}

```
101010 | 110
-110     111₂
 01001
 - 110
  00110
    110
    000
```

Resultado:

111_2

ACTIVIDADES VI

• 31: Investiga el operador XOR y escribe su tabla de verdad.

El operador XOR (también conocido como "OR exclusivo"). Representa una operación que devuelve verdadero (1) cuando el número de entradas verdaderas (1) es impar, y falso (0) cuando el número de entradas verdaderas es par.

INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

• 32: Investiga el operador NAND y escribe su tabla de verdad.

El operador NAND (NO-AND). Es el complemento de la operación lógica AND. Mientras que el operador AND devuelve verdadero (1) solo cuando todas las entradas son verdaderas (1), el operador NAND devuelve verdadero (1) en todos los casos excepto cuando todas las entradas son verdaderas (1).

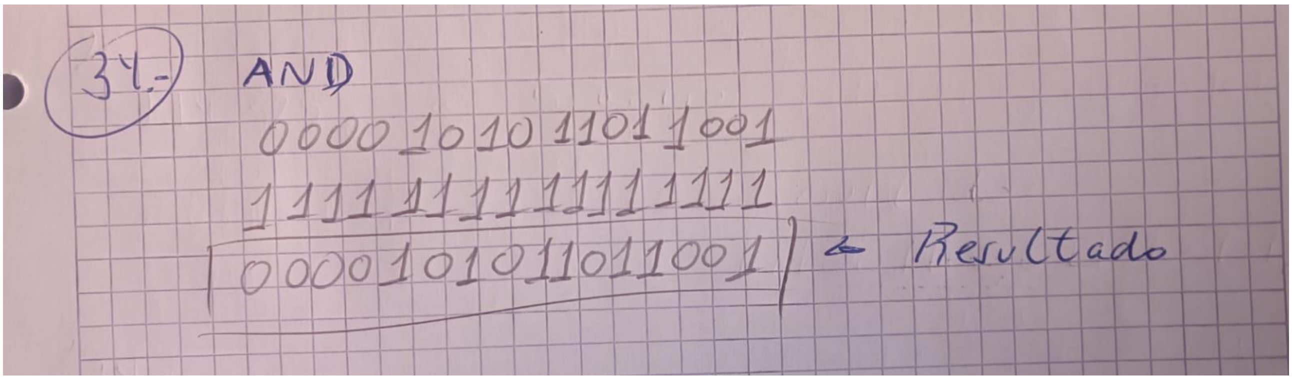
Input A	Input B		Output Q
0	0		1
0	1		1
1	0		1
1	1		0

• 33: Investiga el operador NOR y escribe su tabla de verdad.

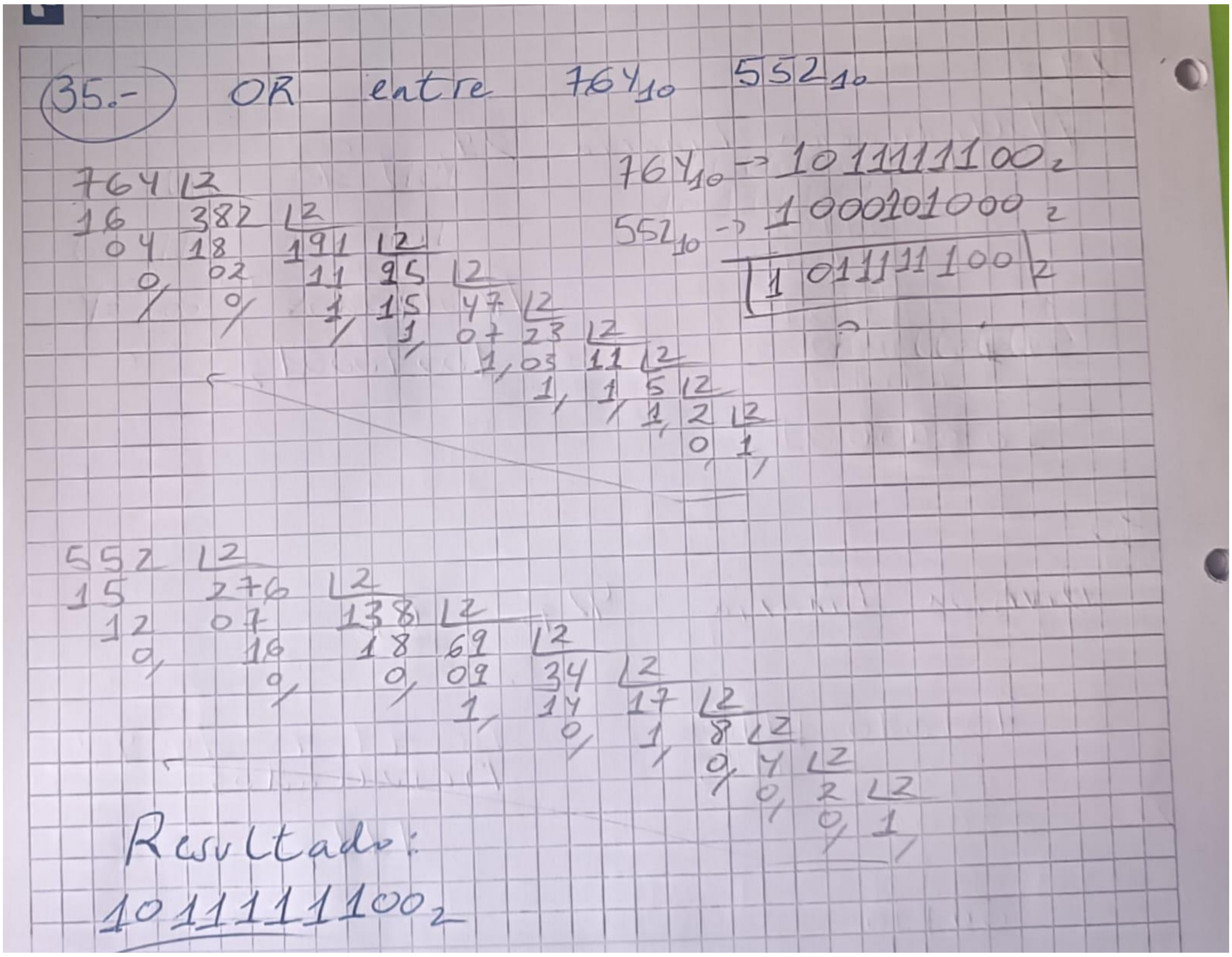
El operador NOR devuelve verdadero (1) cuando ninguna de sus entradas es verdadera (1), es decir, es el complemento de la operación lógica OR.

INPUT		OUTPUT
A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

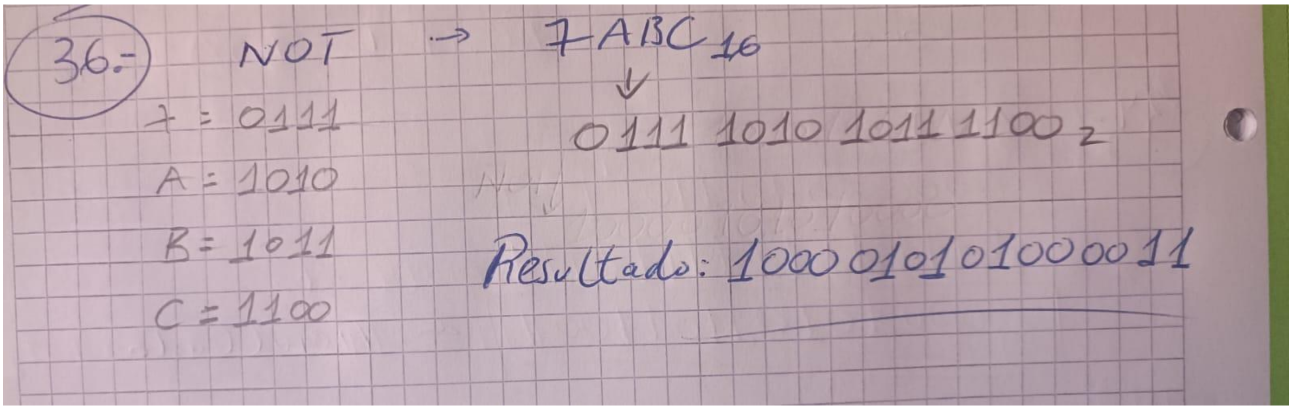
• 34: Realiza un AND lógico entre los números binarios 0000101011011001 y 1111111111111111.



• 35: Realiza un OR lógico entre los números decimales 764 y 552.



• 36: Realiza un NOT lógico del número hexadecimal 7ABC.



ACTIVIDADES VII

• 37: Investiga cuántos números se pueden representar con n bits usando la representación módulo y signo. ¿Cuántas representaciones tendremos para el 0?

Desde -(2^n-1 -1) a +(2^n-1 -1), donde el 0 tiene 2 formas distintas de representación.

- Si el bit de signo es 0 (positivo) y todos los demás bits son 0, entonces representaría el número positivo cero (0).
- Si el bit de signo es 1 (negativo) y todos los demás bits son 0, entonces representaría el número negativo cero (-0).

• 38: Expresa en módulo y signo usando una palabra de 8 bits (cuando se pueda) los siguientes números

a)-128 b) 96 c) -125 d) 0

38.- Módulo y signo usando palabra de 8 bits

a) -128_{10}

128 | 2
08 64 | 2
0 04 32 | 2
0 12 16 | 2
0 0 8 | 2
0 4 | 2
0 2 | 2
0 1

Respuesta:

Signo	Magnitud
1	0000000_2

b) 96_{10}

01100000_2

96 | 2
16 48 | 2
0 08 24 | 2
0 04 12 | 2
0 0 6 | 2
0 3 | 2
1 1

Respuesta:

Signo	Magnitud
0	1100000_2

c) -125_{10}

1111101_2

125 | 2
05 62 | 2
1 02 31 | 2
0 11 15 | 2
1 1 7 | 2
1 3 | 2
1 1

Respuesta:

Signo	Magnitud
1	1111101_2

d) 0

El valor absoluto de 0 en binario en una palabra de 8 bits son todos los bits en 0: 00000000. Indicando el primer bit el signo.

• 39: Investiga cuántos números se pueden representar con n bits usando la representación Complemento a 1. ¿Cuántas representaciones tendremos para el 0?

Un sistema de numeración de complementos de n-bit sólo puede representar enteros en el rango $[-2^{n-1}+1, 2^{n-1}-1]$.
Para el 0 tenemos dos representaciones posibles: una con todos los bits iguales a 0 y otra con todos los bits iguales a 1.

• 40: Expresa en Complemento a 1 usando una palabra de 8 bits los siguientes números
a)-124 b) 56 c) -55 d) 0

40.- Complemento a 1, usando palabra de 8 bits.

a) −124

124 |2
04 62 |2
0 02 31 |2
0 11 15 |2
1 1 7 |2
1 3 |2
1 1

$124_{10} \rightarrow 01111100_{c1}$

$-124_{10} \rightarrow 10000011_{c1}$

Resultado

10000011_{c1}

b) 56

56 |2
16 28 |2
0 08 14 |2
0 0 7 |2
1 3 |2
1 1

00111000_{c1}

Resultado

00111000_{c1}

c) −55

55 |2
15 27 |2
1 07 13 |2
1 1 6 |2
0 3 |2
1 1

$55 \rightarrow 00110111_{c1}$

$-55 \rightarrow 11001000_{c1}$

Resultado:

11001000_{c1}

d) 0

00000000_{c1}

↓

1000000_{c1}

Resultado

00000000_{c1}

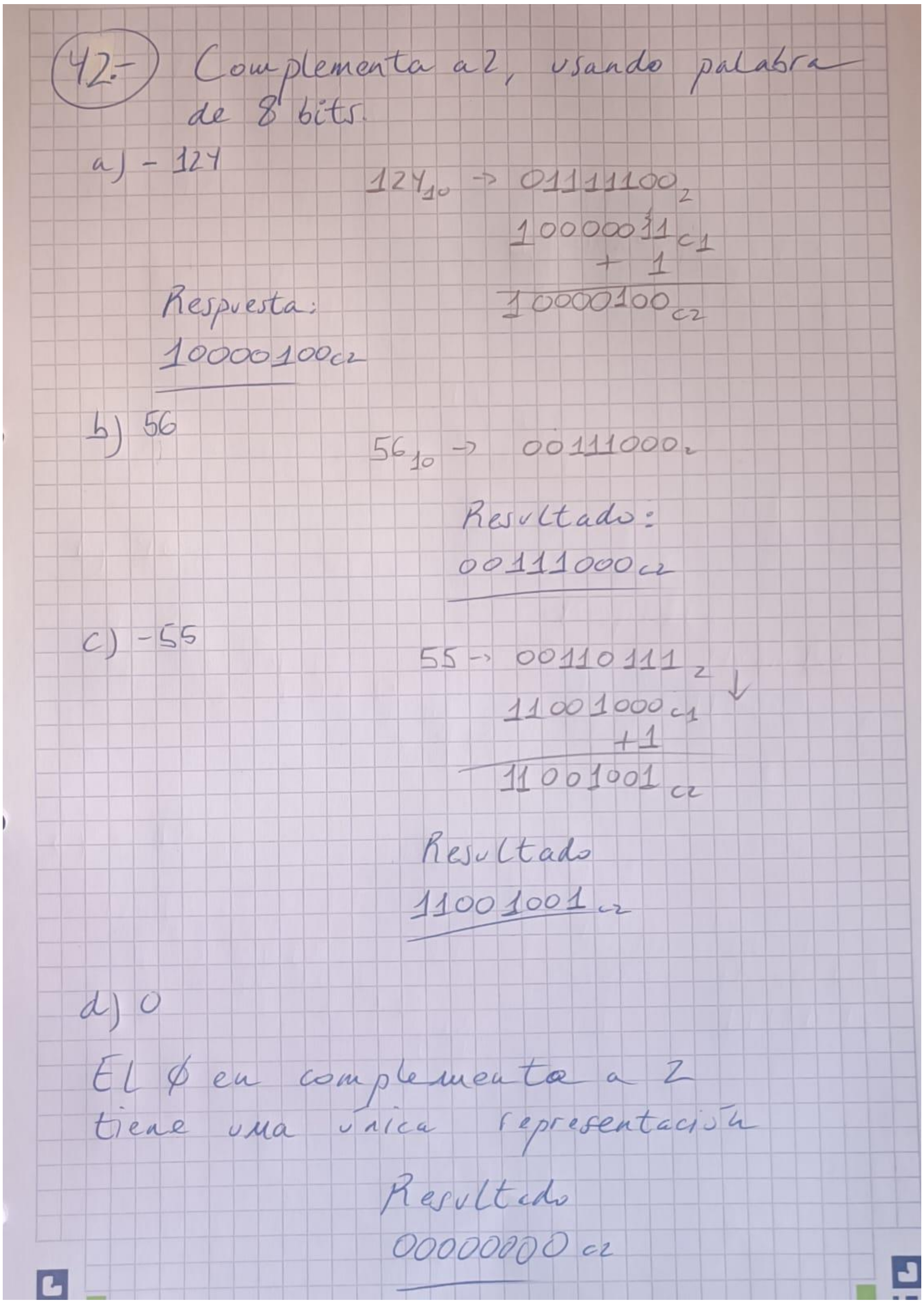
10000000_{c1}

ACTIVIDADES VIII

• 41: Investiga cuántos números se pueden representar con n bits usando la representación Complemento a 2. ¿Cuántas representaciones tendremos para el 0?

El total de números positivos será de 2^{n-1} -1 y el de negativos 2^{n-1}.
Se consigue una única representación del número 0.

• 42: Expresa en Complemento a 2 usando una palabra de 8 bits los siguientes números
a)-124 b) 56 c) -55 d) 0



42.- Complemento a 2, usando palabra de 8 bits.

a) -124

$124_{10} \rightarrow 01111100_2$

10000011_{c1}
$+ 1$
10000100_{c2}

Respuesta:
10000100_{c2}

b) 56

$56_{10} \rightarrow 00111000_2$

Resultado:
00111000_{c2}

c) -55

$55 \rightarrow 00110111_2$

11001000_{c1}
$+1$
11001001_{c2}

Resultado
11001001_{c2}

d) 0

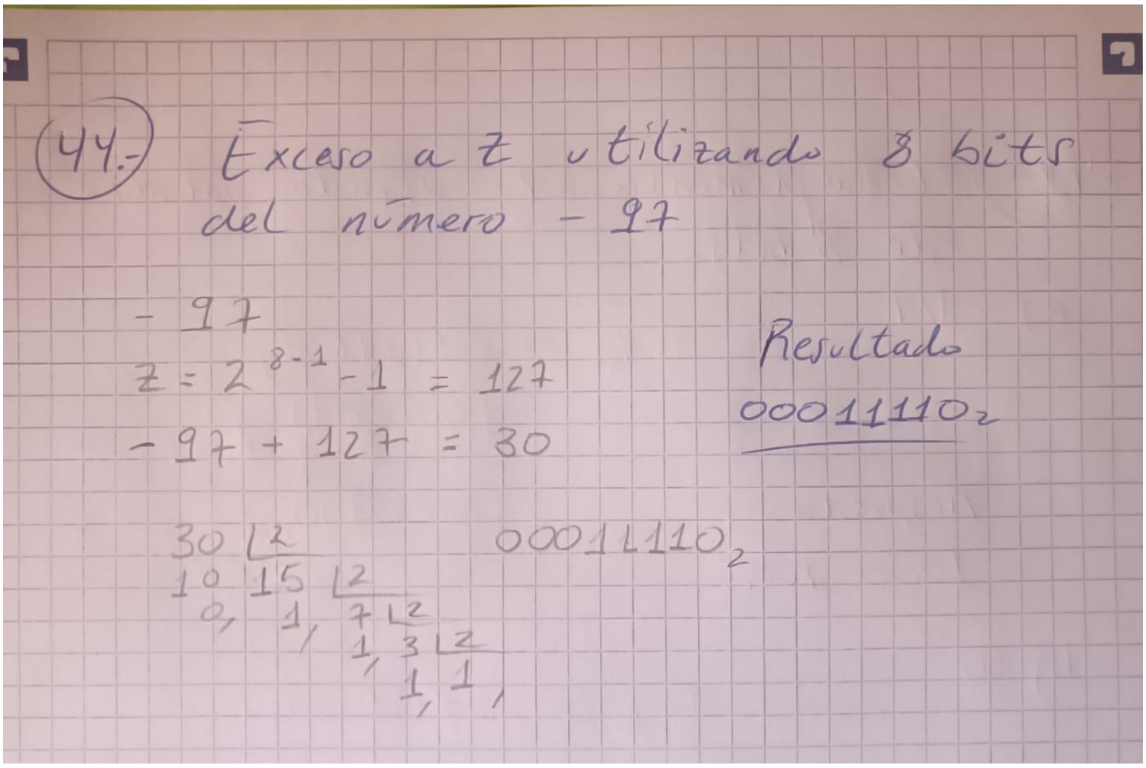
El ∅ en complemento a 2 tiene una única representación

Resultado
00000000_{c2}

• 43: Investiga cuántos números se pueden representar con n bits usando la representación Exceso a Z.

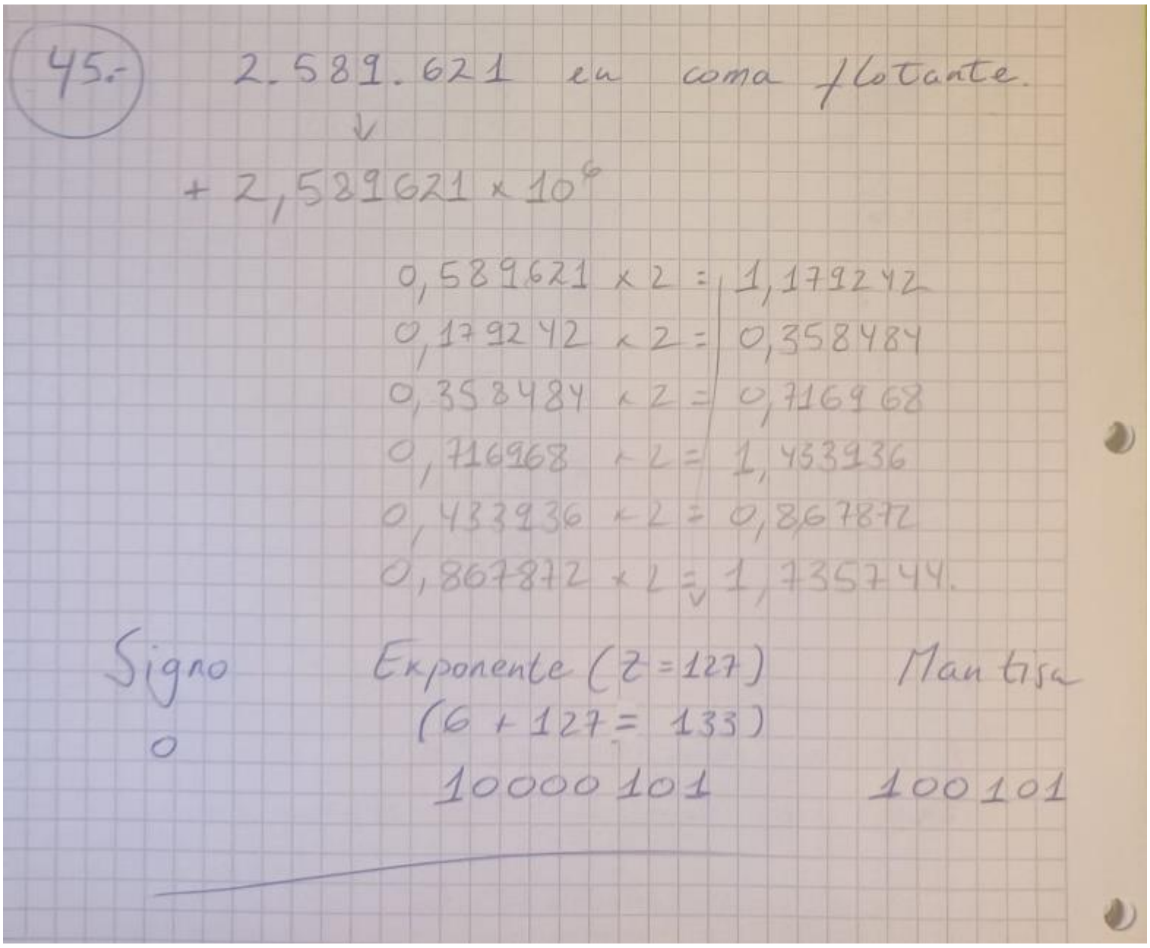
El número más pequeño que se puede representar es 0, y el número más grande 2^{n-1} -1.

• 44: Representa en Exceso a Z utilizando 8 bits (n=8) el número -97.



ACTIVIDADES IX

• 45: Representa el número 2.589.621 en coma flotante.



• 46: Investiga cuándo se produce el desbordamiento también conocido como overflow.

Se produce cuando el número resultante tiene más bits de los que se pueden representar en el espacio de memoria asignado.

• 47: ¿Cómo representa el código ASCII la letra “A” en binario? ¿Y el carácter “#”?

Letra A = 0100 0001
Carácter # = 0010 0011

• 48: ¿Cómo se codificaría la palabra “Verano” en código ASCII?

	Decimal	Binario
V	86	0101 0110
e	101	0110 0101
r	114	0111 0010
a	97	0110 0001
n	110	0110 1110
o	111	0110 1111

• 49: Investiga si se puede codificar la palabra “España” en código ASCII. ¿Y en ASCII extendido?

No se puede codificar en ASCII ya que solo incluye caracteres que son comunes en inglés y no incluye letras acentuadas ni caracteres específicos del español como la letra “ñ”.

En ASCII extendido si se puede codificar:

	Decimal	Binario
E	69	0100 0101
s	115	0111 0011
p	112	0111 0000
a	97	0110 0001
ñ	241	1111 0001
a	97	0110 0001

• 50: Escribe tu nombre de pila en ASCII extendido.

	Decimal	Binario
R	82	0101 0010
o	111	0110 1111
d	100	0110 0100
r	114	0111 0010
i	105	0110 1001
g	103	0110 0111
o	111	0110 1111