# Ejercicios de representación de números enteros.

Si no se especifica lo contrario, debes asumir que los números negativos se representan en complemento a 2 (Ca2).

Si no se especifica lo contrario, debes asumir que el valor de un número se expresa en decimal.

1. Representa en módulo y signo (MS ó SM), Ca1 y Ca2 los valores 25, -25, 112, -112, 0, -0

**MS**

Los números positivos se ponen igual. En los negativos se tiene que poner un 1 en su MSB (Bit más significante, representado en negrita)

25: **0** 0 0 1 1 0 0 1

-25: **1** 0 0 1 1 0 0 1

112: **0** 1 1 1 0 0 0 0

-112: **1** 1 1 1 0 0 0 0

0: **0** 0 0 0 0 0 0 0

-0: **1** 0 0 0 0 0 0 0

**Ca1**

Los números positivos se ponen de igual manera, mientras que para los negativos se tienen que invertir los números, es decir, los 0 pasan a ser 1 y viceversa.

25: **0** 0 0 1 1 0 0 1

-25: **1** 1 1 0 0 1 1 0

112: **0** 1 1 1 0 0 0 0

-112: **1** 0 0 0 1 1 1 1

0: **0** 0 0 0 0 0 0 0

-0: **1** 1 1 1 1 1 1 1

**Ca2**

Los números positivos se representan como siempre, pero para los negativos hay que hacer una primera conversión a Ca1 para después sumarle a este un bit.

25: **0** 0 0 1 1 0 0 1

-25: **1** 1 1 0 0 1 1 0 <- Paso a Ca1

+ 1

**1** 1 1 0 0 1 1 1

112: **0** 1 1 1 0 0 0 0

-112: **1** 0 0 0 1 1 1 1 <- Paso a Ca1

+ 1

**1** 0 0 1 0 0 0 0

0: **0** 0 0 0 0 0 0 0

-0: **1** 1 1 1 1 1 1 1 <- Paso a Ca1

+ 1

**0** 0 0 0 0 0 0 0 (Sobraría el último acarreo)

1. Representa en hexadecimal y en binario el valor 3600

**Binario**

3600 / 2 = 1800, resto **0**

1800 / 2 = 900, resto **0**

900 / 2 = 450, resto **0**

450 / 2 = 225, resto **0**

225 / 2 = 112, resto **1**

112 / 2 = 56, resto **0**

56 / 2 = 28, resto **0**

28 / 2 = 14, resto **0**

14 / 2 = 7, resto **0**

7 / 2 = 3, resto **1**

3 / 2 = **1**, resto **1**

Resultado: 1 1 1 0 0 0 0 1 0 0 0 0

**Hexadecimal**

Uso el binario anterior, y cojo de derecha a izquierda grupos de 4 bits

1 1 1 0 0 0 0 1 0 0 0 0

C 1 0

Resultado: C10

1. Representa en hexadecimal y en binario el valor -250

Primero se pasa a binario como un número más:

250 / 2 = 125, resto **0**

125 / 2 = 62, resto **1**

62 / 2 = 31, resto **0**

31 / 2 = 15, resto **1**

15 / 2 = 7, resto **1**

7 / 2 = 3, resto **1**

3 / 2 = **1**, resto **1**

**Binario:** 1 1 1 1 1 0 1 0

En caso de que queramos pasarlo a negativo usando Ca2, necesitaremos usar 16 bits, ya que con 8 bits nos quedaríamos cortos y ocurriría ***overflow***.

250: **0** 0 0 0 0 0 0 0 1 1 1 1 1 0 1 0

-250: **1** 1 1 1 1 1 1 1 0 0 0 0 0 1 0 1 <- Paso a Ca1

+ 1

1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 0

Ahora se puede pasar a hexadecimal, que sería cogiendo grupos de 4 empezando por la derecha.

1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 0

F F 0 6

**Hexadecimal:** FF06

1. Trabajando con variables de tamaño byte con signo, expresa en binario, decimal y en hexadecimal el resultado de 8D+2A

Paso los números a binario

8D → 1 0 0 0 1 1 0 1

2A → 0 0 1 0 1 0 1 0 +

1 0 1 1 0 1 1 1

**Binario**: 1 0 1 1 0 1 1 1

**Hexadecimal**: 1 0 1 1 1 0 0 0

B 8

**Decimal**: 23 + 24 + 25 + 27 = 184

1. Trabajando con variables de tamaño byte con signo, expresa en binario, decimal y en hexadecimal el resultado de 01010101 + 00111010

0 1 0 1 0 1 0 1

0 0 1 1 1 0 1 0 +

1 0 0 0 1 1 1 1

**Binario:** 1 0 0 0 1 1 1 1

**Hexadecimal:** 1 0 0 0 1 1 1 1

8 F

**Decimal:** 20 + 21 + 22 + 23 + 27 = 143