Práctica 1: Representación de la información

Números enteros

Organización del Computador I
DC - UBA

Verano 2018

¿Cómo se representan los números?

Cada número se puede representar de **varias** maneras.

Por ejemplo: el vigésimo elemento de los números naturales sin el cero.

- Si se escribiese en la antigua Roma hubiese sido:

¿Cómo se representan los números?

Cada número se puede representar de **varias** maneras.

Por ejemplo: el vigésimo elemento de los números naturales sin el cero.

- Si se escribiese en la antigua Roma hubiese sido:

$$\text{XX}$$

- Si se escribiese acá mismo, sería:

¿Cómo se representan los números?

Cada número se puede representar de **varias** maneras.

Por ejemplo: el vigésimo elemento de los números naturales sin el cero.

- Si se escribiese en la antigua Roma hubiese sido:

  XX

- Si se escribiese acá mismo, sería:

  20

- Aunque... también podría haber sido:

¿Cómo se representan los números?

Cada número se puede representar de **varias** maneras.

Por ejemplo: el vigésimo elemento de los números naturales sin el cero.

- Si se escribiese en la antigua Roma hubiese sido:

$$\mathrm{XX}$$

- Si se escribiese acá mismo, sería:

$$20$$

- Aunque... también podría haber sido:

$$10100$$

Sistemas no posicionales: Los porotos

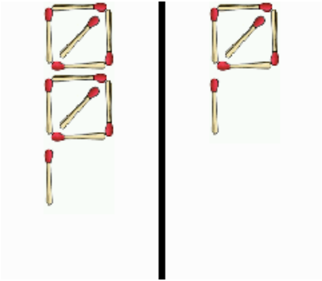
Para representar un número natural podemos usar:

En este caso, representan el número 27.

¿Importa el orden de los porotos?

Sistemas no posicionales: Tanteando en el truco

Cuando jugamos al truco, anotamos los puntos de una manera particular:

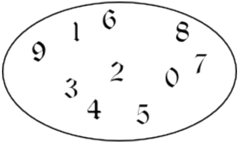


Aquí vemos que un equipo tiene 11 puntos y el otro 6.

¿Importa el orden de los palitos? Sólo los juntamos para contar más rápido...

Sistemas posicionales: Sistema decimal

Un sistema decimal utiliza un conjunto que tiene 10 símbolos. Usualmente utilizamos estos:

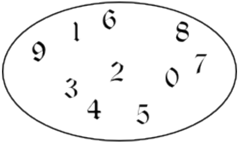


¿Qué número representa la siguiente tira de símbolos?

478

Sistemas posicionales: Sistema decimal

Un sistema decimal utiliza un conjunto que tiene 10 símbolos.
Usualmente utilizamos estos:



¿Qué número representa la siguiente tira de símbolos?

478

¿Podría ser $4 + 7 + 8 =$ *diecinueve*?

¿Qué es una base?

- El sistema *decimal* utiliza un conjunto de 10 símbolos para representar los números.

¿Qué es una base?

- El sistema *decimal* utiliza un conjunto de 10 símbolos para representar los números.
- Así, en cada numeral, la posición de cada símbolo está relacionada con una potencia de 10.

¿Qué es una base?

- El sistema *decimal* utiliza un conjunto de 10 símbolos para representar los números.
- Así, en cada numeral, la posición de cada símbolo está relacionada con una potencia de 10.

¿Y si en vez de diez símbolos tuviéramos otra cantidad?

¿Qué es una base?

- El sistema *decimal* utiliza un conjunto de 10 símbolos para representar los números.
- Así, en cada numeral, la posición de cada símbolo está relacionada con una potencia de 10.

¿Y si en vez de diez símbolos tuviéramos otra cantidad?

A esa cantidad la llamamos **base**

Representaciones e interpretaciones de números.

- ¿Qué números pueden representar las siguientes cadenas?
  10
  478
  2011
  ORGA1

Representaciones e interpretaciones de números.

- ¿Qué números pueden representar las siguientes cadenas?
  10
  478
  2011
  ORGA1
- Dada la representación de un número, ¿puedo saber qué base se está utilizando?

Bases.

- En base 2, usamos los símbolos 0 y 1 y escribimos los naturales: 0, 1, 10, 11, 100, 101, 110...
- En base 3, usamos los símbolos 0, 1 y 2 y escribimos los naturales: 0, 1, 2, 10, 11, 12, 20...
- ...y así...

Bases más comunes

Base	Símbolos usados
2 (binario)	0, 1
8 (octal)	0, 1, 2, 3, 4, 5, 6, 7
10 (decimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
16 (hexadecimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Bases.

- En base 2, usamos los símbolos 0 y 1 y escribimos los naturales: 0, 1, 10, 11, 100, 101, 110...
- En base 3, usamos los símbolos 0, 1 y 2 y escribimos los naturales: 0, 1, 2, 10, 11, 12, 20...
- ...y así...

Bases más comunes

Base	Símbolos usados
2 (binario)	0, 1
8 (octal)	0, 1, 2, 3, 4, 5, 6, 7
10 (decimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
16 (hexadecimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

¿Es posible cambiar un número de una base a otra?

Bases.

- En base 2, usamos los símbolos 0 y 1 y escribimos los naturales: 0, 1, 10, 11, 100, 101, 110...
- En base 3, usamos los símbolos 0, 1 y 2 y escribimos los naturales: 0, 1, 2, 10, 11, 12, 20...
- ...y así...

Bases más comunes

Base	Símbolos usados
2 (binario)	0, 1
8 (octal)	0, 1, 2, 3, 4, 5, 6, 7
10 (decimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
16 (hexadecimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

¿Es posible cambiar un número de una base a otra?

¿Cómo escribirían un programa que lo haga?

Cambios de bases: Teorema de la división.

Teorema

Sean $a \in \mathbb{Z}$ y $b \in \mathbb{N}$.
Existen $q, r \in \mathbb{Z}$ con $0 \leq r < b$
tales que $a = b \times q + r$
Además, q y r son únicos.

Cambios de bases: Teorema de la división.

Teorema

Sean $a \in \mathbb{Z}$ y $b \in \mathbb{N}$.
Existen $q, r \in \mathbb{Z}$ con $0 \leq r < b$
tales que $a = b \times q + r$
Además, q y r son únicos.

Ejercitación

Escribir los siguientes números en binario, octal y hexadecimal.

- diez
- quinientos doce

Sumando...

En decimal	En base 3
$\begin{array}{r} 8\ 4\ 5 \\ +\ 3\ 4\ 2 \\ \hline 1\ 1\ 8\ 7 \end{array}$	$\begin{array}{r} 2\ 1\ 2 \\ +\ 1\ 0\ 1 \\ \hline 1\ 0\ 2\ 0 \end{array}$

Multiplicando...

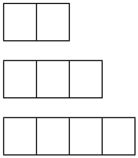
En base 3

$$
\begin{array}{r}
1\ 2 \\
\times\ 2\ 1 \\
\hline
1\ 2 \\
1\ 0\ 1\ \text{-} \\
\hline
1\ 0\ 2\ 2
\end{array}
$$

Tiras de símbolos

Si sólo podemos escribir tiras de símbolos de longitud fija:

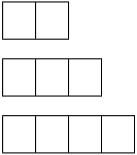
- ¿cuántos números podemos representar?
- ¿de qué depende?



Tiras de símbolos

Si sólo podemos escribir tiras de símbolos de longitud fija:

- ¿cuántos números podemos representar?
- ¿de qué depende?



Al operar con precisión fija, podemos tener **overflow**

Ocurre overflow cuando el resultado de una operación necesita una tira de símbolos más grande que la disponible

Hasta aca: números naturales

Por ahora sabemos...

- Leer naturales
- Escribir naturales
- Cambiarlos de base
- Operar con naturales (sumarlos, multiplicarlos)

¿Y qué hacemos con los enteros?

Codificando números enteros en base binaria

- Sin signo

Codificando números enteros en base binaria

- Sin signo
  solo sirve para positivos.

Codificando números enteros en base binaria

- Sin signo
  solo sirve para positivos.
- Signo + Magnitud

Codificando números enteros en base binaria

- Sin signo
  solo sirve para positivos.
- Signo + Magnitud
  se usa un bit para indicar el signo

Codificando números enteros en base binaria

- Sin signo
  solo sirve para positivos.
- Signo + Magnitud
  se usa un bit para indicar el signo
- Complemento a 2

Codificando números enteros en base binaria

- Sin signo
  solo sirve para positivos.
- Signo + Magnitud
  se usa un bit para indicar el signo
- Complemento a 2
  los positivos se representan igual, un número negativo n como $2^k + n$

Codificando números enteros en base binaria

- Sin signo
  solo sirve para positivos.
- Signo + Magnitud
  se usa un bit para indicar el signo
- Complemento a 2
  los positivos se representan igual, un número negativo n como $2^k + n$
- Exceso m

Codificando números enteros en base binaria

- Sin signo
  solo sirve para positivos.
- Signo + Magnitud
  se usa un bit para indicar el signo
- Complemento a 2
  los positivos se representan igual, un número negativo n como $2^k + n$
- Exceso m
  represento n como $m + n$

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	?		
-2			
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011		
-2			
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	?	
-2			
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	
-2			
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	?
-2			
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2			
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	?		
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010		
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	?	
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	?
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	1101
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	1101
-8	?		

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	1101
-8	OVERFLOW		

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	1101
-8	OVERFLOW	?	

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	1101
-8	OVERFLOW	1000	

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	1101
-8	OVERFLOW	1000	?

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	1101
-8	OVERFLOW	1000	0111

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	?		
-2			
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011		
-2			
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	?	
-2			
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	
-2			
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	?
-2			
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2			
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	?		
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010		
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	?	
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	?
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8	?		

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8	1000 1000		

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8	1000 1000	?	

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8	1000 1000	1111 1000	

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8	1000 1000	1111 1000	?

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8	1000 1000	1111 1000	0000 0111

Codificando...(con más bits)

Similitudes entre 4 y 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8	1000 1000	1111 1000	0000 0111

Codificando...(con más bits)

Similitudes entre 4 y 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8	1000 1000	1111 1000	0000 0111

Extendiendo la cantidad de bits de precisión:

- Signo + Magnitud: Se extiende con 0's, pero el bit más significativo se mantiene indicando el signo.
- Complemento a 2: Se extiende con el valor del bit más significativo.
- Complemento a m: Se extiende siempre con 0's.

Resumiendo: Codificando números enteros en numerales binarios

Sin Signo

Solo sirve para los positivos.
numeral $\rightarrow$ número que representa

$1111 \rightarrow 15_{10}$	$0111 \rightarrow 7_{10}$
$1110 \rightarrow 14_{10}$	$0110 \rightarrow 6_{10}$
$1101 \rightarrow 13_{10}$	$0101 \rightarrow 5_{10}$
$1100 \rightarrow 12_{10}$	$0100 \rightarrow 4_{10}$
$1011 \rightarrow 11_{10}$	$0011 \rightarrow 3_{10}$
$1010 \rightarrow 10_{10}$	$0010 \rightarrow 2_{10}$
$1001 \rightarrow 9_{10}$	$0001 \rightarrow 1_{10}$
$1000 \rightarrow 8_{10}$	$0000 \rightarrow 0_{10}$

Para los numerales de 4 *bits*.

Resumiendo: Codificando números enteros en numerales binarios

Signo+Magnitud

El primer bit es el signo, los demás son el *significado* (la magnitud del número en valor absoluto).

numeral → número que representa

$1111 \rightarrow -7_{10}$	$0111 \rightarrow 7_{10}$
$1110 \rightarrow -6_{10}$	$0110 \rightarrow 6_{10}$
$1101 \rightarrow -5_{10}$	$0101 \rightarrow 5_{10}$
$1100 \rightarrow -4_{10}$	$0100 \rightarrow 4_{10}$
$1011 \rightarrow -3_{10}$	$0011 \rightarrow 3_{10}$
$1010 \rightarrow -2_{10}$	$0010 \rightarrow 2_{10}$
$1001 \rightarrow -1_{10}$	$0001 \rightarrow 1_{10}$
$1000 \rightarrow -0_{10}$	$0000 \rightarrow 0_{10}$

Para los numerales de 4 *bits*.

Resumiendo: Codificando números enteros en numerales binarios

Complemento a dos

Los numerales que representa positivos son iguales a los anteriores
Para los negativos, dado un n negativo se representan escribiendo
$2^k + n$ en notación sin signo

cuentas $\rightarrow$ numeral $\rightarrow$ número que representa

$2^4 + (-1) = 15 \rightarrow 1111 \rightarrow -1_{10}$	$0111 \rightarrow 7_{10}$
$2^4 + (-2) = 14 \rightarrow 1110 \rightarrow -2_{10}$	$0110 \rightarrow 6_{10}$
$2^4 + (-3) = 13 \rightarrow 1101 \rightarrow -3_{10}$	$0101 \rightarrow 5_{10}$
$2^4 + (-4) = 12 \rightarrow 1100 \rightarrow -4_{10}$	$0100 \rightarrow 4_{10}$
$2^4 + (-5) = 11 \rightarrow 1011 \rightarrow -5_{10}$	$0011 \rightarrow 3_{10}$
$2^4 + (-6) = 10 \rightarrow 1010 \rightarrow -6_{10}$	$0010 \rightarrow 2_{10}$
$2^4 + (-7) = 9 \rightarrow 1001 \rightarrow -7_{10}$	$0001 \rightarrow 1_{10}$
$2^4 + (-8) = 8 \rightarrow 1000 \rightarrow -8_{10}$	$0000 \rightarrow 0_{10}$

Para los numerales de 4 *bits*.

Codificando números enteros en numerales binarios

Exceso a m

El número n se representa como $m + n$

cuentas $\rightarrow$ numeral $\rightarrow$ número que representa

$5 + (10) = 15 \rightarrow$	$1111 \rightarrow$	10_{10}	$5 + (2) = 7 \rightarrow$	$0111 \rightarrow$	2_{10}
$5 + (9) = 14 \rightarrow$	$1110 \rightarrow$	9_{10}	$5 + (1) = 6 \rightarrow$	$0110 \rightarrow$	1_{10}
$5 + (8) = 13 \rightarrow$	$1101 \rightarrow$	8_{10}	$5 + (0) = 5 \rightarrow$	$0101 \rightarrow$	0_{10}
$5 + (7) = 12 \rightarrow$	$1100 \rightarrow$	7_{10}	$5 + (-1) = 4 \rightarrow$	$0100 \rightarrow$	-1_{10}
$5 + (6) = 11 \rightarrow$	$1011 \rightarrow$	6_{10}	$5 + (-2) = 3 \rightarrow$	$0011 \rightarrow$	-2_{10}
$5 + (5) = 10 \rightarrow$	$1010 \rightarrow$	5_{10}	$5 + (-3) = 2 \rightarrow$	$0010 \rightarrow$	-3_{10}
$5 + (4) = 9 \rightarrow$	$1001 \rightarrow$	4_{10}	$5 + (-4) = 1 \rightarrow$	$0001 \rightarrow$	-4_{10}
$5 + (3) = 8 \rightarrow$	$1000 \rightarrow$	3_{10}	$5 + (-5) = 0 \rightarrow$	$0000 \rightarrow$	-5_{10}

Para los numerales de 4 *bits* en exceso 5.

Mañana

- Codificación de números con coma
- Codificación de caracteres