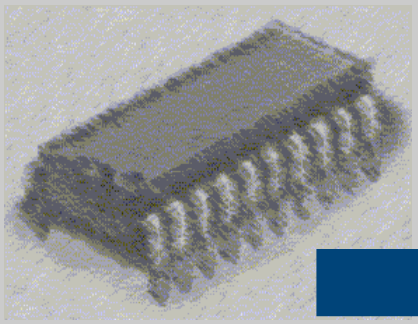
Enunciados de problemas

Tema 2. Sistemas de representación de la información



Estructura de Computadores

I. T. Informática de Gestión / Sistemas

Curso 2008-2009

Base teórica

El computador solamente entiende de bits, por lo que toda la información que empleamos en nuestra vida diaria tiene que ser convertida a bits para que el ordenador lo entienda. Así, los números, las letras y los caracteres numéricos tienen que ser convertidos a bits.

Bases de numeración

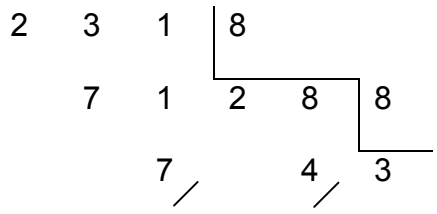
Los números se pueden encontrar en diferentes bases de numeración. Para poder cambiar entre bases, se deben aplicar las reglas de Horner.

Regla de Horner para números enteros

Es un algoritmo muy sencillo. Consiste en lo siguiente:

1. Dividir el número a convertir por la base a la que se desea la conversión y quedarse con el resto.
2. Si el cociente es mayor o igual que la base a convertir repetir el paso 1.
3. Si el cociente es menor que la base a la que deseamos convertir el número, ya hemos terminado. El número convertido se forma tomando el cociente como primera cifra del número y luego concatenando los restos, desde el último hasta el primero.

Ejemplo de aplicación: convertir a base 8 el número decimal 231



El número convertido será por tanto el $347_{(8}$

Regla de Horner para números fraccionarios

Es un algoritmo igualmente sencillo. La diferencia con el anterior es que se multiplica por la base a convertir en lugar de dividir por ella.

En el caso de base dos, el algoritmo sería

1. Tomar la parte fraccionaria del número y multiplicarla por dos
2. Tomar la parte entera como primera cifra del número y volver al paso 1

El algoritmo termina cuando la conversión ha sido exacta, la parte fraccionaria es cero, o si se han completado todos los bits disponibles para el número

Ejemplo de aplicación. Convertir el número 0,39 a base dos limitándolo únicamente a 8 bits.

0,39 x 2 = 0,78 → nos quedamos con la parte entera 0

0,78 x 2 = 1,56 → nos quedamos con la parte entera 1

0,56 x 2 = 1,12 → nos quedamos con la parte entera 1

0,12 x 2 = 0,24 → nos quedamos con la parte entera 0

0,24 x 2 = 0,48 → nos quedamos con la parte entera 0

0,48 x 2 = 0,96 → nos quedamos con la parte entera 0

0,96 x 2 = 1,92 → nos quedamos con la parte entera 1

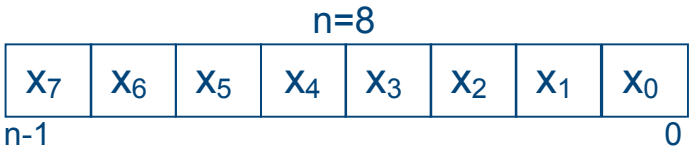
0,92 x 2 = 1,84 → nos quedamos con la parte entera 1

El número convertido sería el 0,01100011 …. Vemos que se podría seguir obteniendo cifras por lo que la conversión no es exacta.

Representaciones numéricas

Los números pueden ser enteros o fraccionarios, sin embargo, el computador únicamente entiende de bits y requiere que los números con los que trabajamos habitualmente, se conviertan en cadenas de ceros y de unos que pueda entender el computador.

Como el sistema decimal en el que nos movemos y el binario son sistemas posicionales, es fácil convertir de unos a otros mediante la regla de Horner.



Tenemos, dos tipos de representaciones numéricas: coma fija (números enteros) y coma flotante (números fraccionarios).

Sistemas de numeración en coma fija

Sirven para representar números enteros. Dependiendo del sistema de representación se tendrán números con signo o números sin signo.

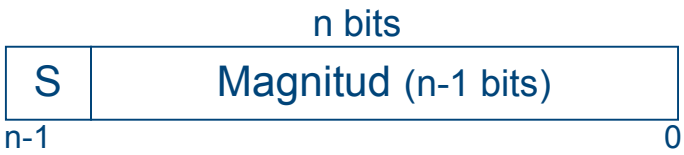
1.- Binario puro (base dos, son enteros sin signo)

Simplemente se convierte el número entero en decimal a base dos mediante la regla de Horner. Dependiendo del número de bits que tengamos para representar el número, se tendrá un rango u otro de representación. En el caso de binario puro es [0, 2n - 1]

El valor del número se calculará como:

$$Valor = \sum_{i=0}^{n-1} 2^i \cdot x_i$$

2.- Signo-magnitud (con signo)



Es casi igual que la representación anterior salvo porque admite números positivos y negativos. Para ello, el bit más significativo, es un bit de signo a parte que indicará si el número es positivo (0) o si es negativo (1).

La magnitud se representa en binario puro.

El rango, en este caso, comprenderá números positivos y negativos, dependiendo del valor del bit más significativo:

- Negativos: $[-(2^{n-1} - 1), -0]$
- Positivos: $[0, (2^{n-1} - 1)]$ (rango simétrico)

El valor del número en decimal se calculará:

$$Valor = \begin{cases} +\sum_{i=0}^{n-2} 2^i \cdot x_i & \text{si } x_{n-1} = 0 \\ -\sum_{i=0}^{n-2} 2^i \cdot x_i & \text{si } x_{n-1} = 1 \end{cases}$$

3.- Complemento a 1 (con signo)

Al igual que en el caso anterior, los números positivos comienzan por cero y los negativos por uno. La diferencia es que ahora, el bit más significativo forma parte del valor del número.

La manera de representar los positivos es en binario puro y haciendo que el bit más significativo sea un cero. En el caso de los números negativos, se debe calcular el complemento a 1 del número.

El complemento a 1 de A será 2^n - A – 1 siendo *n* el número de bits que se tienen para la representación. Otra manera más sencilla consiste en cambiar los unos por ceros y los ceros por unos del número original.

La manera de calcular el valor decimal del número es sencilla en el caso de los positivos, en el de los negativos se requiere volver a realizar el complemento a 1 del número para ver el valor del que se partió.

$$Valor = \begin{cases} +\sum_{i=0}^{n-1} 2^i \cdot x_i & \text{si } x_{n-1} = 0 \\ -Valor(Ca1(número)) & \text{si } x_{n-1} = 1 \end{cases}$$

El rango comprenderá también números positivos y negativos, dependiendo del valor del bit más significativo:

- Negativos: [-(2^{n-1} - 1), -0]
- Positivos: [0, (2^{n-1} - 1)]

4.- Complemento a 2 (con signo)

Al igual que en el caso del complemento a 1, los números positivos comienzan por cero y los negativos por uno.

La manera de representar los positivos es en binario puro y haciendo que el bit más significativo sea un cero. En el caso de los números negativos, se debe calcular el complemento a 2 del número.

El complemento a 2 de A será 2^n - A siendo *n* el número de bits que se tienen para la representación. Otra manera más sencilla consiste en cambiar los unos por ceros y los ceros por unos del número original y sumar uno al resultado. Por último, la manera más rápida consiste en recorrer el número original de derecha a izquierda, dejando el número tal cual hasta que se encuentre el primer uno, y a partir de ese primer uno cambiar los unos por ceros y los ceros por unos.

La manera de calcular el valor decimal del número es sencilla en el caso de los positivos, en el de los negativos se requiere volver a realizar el complemento a 2 del número para ver el valor del que se partió.

$$Valor = \begin{cases} + \sum_{i=0}^{n-1} 2^i \cdot x_i & \text{si } x_{n-1} = 0 \\ -Valor(Ca2(número)) & \text{si } x_{n-1} = 1 \end{cases}$$

El rango comprenderá también números positivos y negativos, dependiendo del valor del bit más significativo:

- Negativos: $[-2^{n-1},-1]$
- Positivos: $[0, (2^{n-1} - 1)]$

5.- Exceso 2^{n-1} (con signo)

Al igual que en complemento a 1 y complemento a 2, el bit que indica el signo es el más significativo y además forma parte del valor del número. La diferencia con los anteriores es que en este formato de representación, si el número comienza por uno es positivo y si es negativo comenzará por cero.

En el caso de exceso 2^{n-1} existe una correspondencia entre los valores en complemento a 2 y exceso 2^{n-1}. Invirtiendo el bit más significativo se pasa de un número en complemento a 2 a exceso 2^{n-1} y al revés.

La manera de trabajar es sumar el exceso al número original y el valor que nos dé, representarlo en binario puro.

La manera de calcular el valor representado en exceso es ver cual es el valor decimal del valor binario representado y restarle el exceso.

El rango es el mismo que en complemento a 2:

- Negativos: $[-2^{n-1}, -1]$
- Positivos: $[0, (2^{n-1} - 1)]$

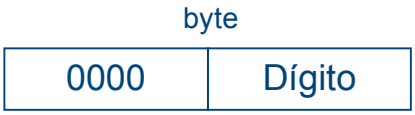
5.- Decimal codificado en binario (BCD)

El sistema consiste en codificar cada cifra del número decimal con los cuatro bits correspondientes en binario.

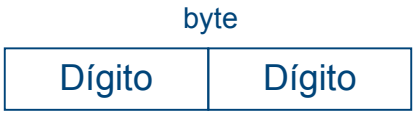
Los dígitos permitidos por tanto serán:

Valor	BCD	Valor	BCD
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

Dependiendo de cómo se agrupen los valores se tendrá BCD desempaquetado (si en un byte solamente se representa una cifra BCD) o empaquetado, si se representan dos cifras BCD por byte.



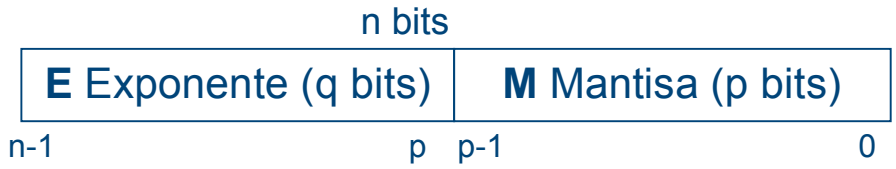
BCD desempaquetado



BCD empaquetado

Sistemas de numeración en coma flotante

Los sistemas de coma flotante sirven para representar números fraccionarios. Para ello, como solamente se dispone de bits para representar los números, parte de los *n* bits del número se emplean para la mantisa y el resto para el exponente del número, adoptando una notación exponencial del número de la forma:



Representación en coma flotante

Por lo que el valor del número representado será:

Valor(x) = valor de la mantisa x $2^{\text{exponente}}$

El exponente siempre se representará en alguna de las notaciones de coma fija anteriores.

La mantisa puede ser entera (ya no se emplea), o fraccionaria.

En el caso de que sea fraccionaria, se tiene que ver si el número se encuentra normalizado o no.

Un número normalizado tiene una única representación y ha eliminado todos los dígitos no significativos a la derecha de la coma. La normalización, por lo tanto, depende del sistema de representación que esté empleando la mantisa:

$$Signo-magnitud \begin{cases} Nümeros\ Positivos: & ,1xxxxx...x \\ Números\ Negativos: & ,1xxxxx...x \end{cases}$$

$$Complemento\ a\ 1 \begin{cases} Nümeros\ Positivos: & ,01xxxxx...x \\ Números\ Negativos: & ,10xxxxx...x \end{cases}$$

$$Complemento\ a\ 2 \begin{cases} Nümeros\ Positivos: & ,01xxxxx...x \\ Números\ Negativos: & ,10xxxxx...x \end{cases}$$

Técnica del bit implícito

Como se ve en las condiciones de normalización anteriores, se pierden en el peor de los casos dos bits para la normalización. Por tanto, la técnica del bit implícito indica que solamente se represente uno de los dos, en el caso de mantisas en complemento a 1 o en complemento a 2, para poder reconstruir el signo del número; y ninguno en el caso de la representación en signo-magnitud dado que se sabe que comienza siembre por uno.

El bit implícito lo emplea internamente la ALU para calcular los valores de los números y los rangos. A efectos prácticos es como si el sistema contase con un bit más para la mantisa.

Las condiciones de normalización con bit implícito quedarían:

$$Signo-magnitud \begin{cases} Nümeros\ Positivos: & ,xxxxx...x \\ Números\ Negativos: & ,xxxxx...x \end{cases}$$

$$Complemento\ a\ 1 \begin{cases} Nümeros\ Positivos: & ,1xxxxx...x \\ Números\ Negativos: & ,0xxxxx...x \end{cases}$$

$$Complemento\ a\ 2 \begin{cases} Nümeros\ Positivos: & ,1xxxxx...x \\ Números\ Negativos: & ,0xxxxx...x \end{cases}$$

Formato IEEE 754 para coma flotante

La organización IEEE ha desarrollado un estándar para la representación de números en coma flotante que ya incorporan la mayoría de los micros actuales. El formato definido es:

- **Exponente:** representado en exceso $2^{q-1} - 1$
- **Mantisa:** representada en signo-magnitud, fraccionaria, normalizada y con la coma situada a la derecha del bit implícito.

- **Simple precisión:**
  - Exponente de 8 bits en exceso 28-1 - 1 = 127
  - Mantisa de 24 bits (1 bit de signo y 23 de magnitud)

Signo	Exponente	Mantisa
1 bit	*8 bits*	*23 bits*

- **Doble precisión**:
  - Exponente de 11 bits en exceso 211-1 - 1 = 1.023
  - Mantisa de 53 bits (1 bit de signo y 52 de magnitud)

Signo	Exponente	Mantisa
1 bit	*11 bits*	*52 bits*

Representaciones alfanuméricas

Sirven para representar los caracteres y las cadenas de caracteres. La manera más extendida es de emplear para la codificación individual de los caracteres la tabla ASCII o el estándar UNICODE.

Para representar las cadenas de caracteres se pueden emplear.

- **Cadenas de longitud fija:**

Se define una longitud máxima para todas las cadenas.

P E P E	A N T O N I O	R O S A

- **Cadenas de longitud variable:**
  - Con carácter separador

*	P	E	P	E	*	A	N	T	O	N	I	O	*	R	O	S	A

  - Con longitud explícita

4	P	E	P	E	7	A	N	T	O	N	I	O	4	R	O	S	A

Representaciones redundantes

Tiene como objetivo el que la información representada sea correcta, y en el caso de que no lo sea detectarla y en algunos casos corregirla.

A los códigos que únicamente detectan los errores se les denomina códigos detectores. A los que además la corrigen se les denomina códigos correctores.

Uno de los códigos detectores más sencillo y empleado es el código de paridad. Por ejemplo el código de paridad par es un bit que se añade a la información. Este bit tomará el valor uno si el número de bits a uno de la información es impar y cero en caso contrario. La idea es que si cuentas el número de unos de la información más el bit de paridad sea un número par.

Como primer código corrector está una adaptación del anterior. Es el código de paridad por bloques. Consiste en proteger varias palabras de información añadiéndoles un bit de paridad horizontal y otro vertical de manera que si fallan en ambos sitios se sepa cual es el bit erróneo y se pueda corregir.

Dentro de los códigos de paridad el más empleado es el código Hamming que minimiza el número de bits requeridos para proteger el número. Los bits de protección de Hamming, se colocan en el lugar de las potencias de dos y protegerán a los bits cuyo subíndice formen parte de la

descomposición binaria del número, así el bit 7 estará protegido por los bits de paridad de Hamming 4, 2 y 1. El bit 5 por el 4 y el 1, etc.

El número de bits necesarios por Hamming los da la inecuación:

$$2p \geq n + p + 1$$

donde:

- n es el número de bits de datos del código
- p es el número de bits de paridad que se añaden

También existen otra serie de códigos redundates como son los códigos polinomiales, muy empleados por los MODEM.

1. Se tiene un ordenador con los siguientes formatos de representación:
   - Números enteros con 16 bits, representados en signo-magnitud: 1 bit para el signo y 15 para la magnitud.
   - Números en coma flotante con las siguientes características:
   - Mantisa fraccionaria expresada en signo-magnitud.
   - El ancho de la mantisa es de 24 bits: 1 bit de signo y 23 de magnitud.
   - La mantisa está normalizada y la coma se sitúa a la izquierda del bit más significativo.
   - El exponente viene representado en exceso 2^{8-1}.

     El ancho del exponente son 8 bits.

Se pide:

a) Calcular el rango de representación para los números enteros.

b) Calcular el rango de representación para los números en coma flotante en el caso de utilizar una mantisa con bit implícito y sin bit implícito:

c) Representar en dicho formato los números:

- 24
- -24
- 24,25 considerando que la mantisa se representa sin bit implícito.
- 24,25 considerando que la mantisa se representa con bit implícito.
- -24,25 considerando que la mantisa se representa sin bit implícito.
- -24,25 considerando que la mantisa se representa con bit implícito.

---

2. Se tiene un ordenador con los siguientes formatos de representación:
   - Números enteros con 24 bits, representados en complemento a 1.

- Números en coma flotante con las siguientes características:
- Mantisa fraccionaria expresada en complemento a 2.
- El ancho de la mantisa es de 24 bits.
- La mantisa está normalizada y la coma se sitúa a la izquierda del bit más significativo.
- El exponente viene representado en exceso 2^{8-1}.
- El ancho del exponente son 8 bits.

Se pide:

a) Calcular el rango de representación para los números enteros.

b) Calcular el rango de representación para los números en coma flotante en el caso de utilizar una mantisa con bit implícito y sin bit implícito.

c) Representar en dicho formato los números:

- 37
- -214
- 14,25 considerando que la mantisa se representa sin bit implícito.
- 14,25 considerando que la mantisa se representa con bit implícito.
- -14,25 considerando que la mantisa se representa sin bit implícito.

3. Se tiene un ordenador con los siguientes formatos de representación:
   - Números enteros con 8 bits, representados en signo-magnitud.
   - Números en coma flotante con las siguientes características:
   - Mantisa fraccionaria expresada en signo-magnitud
   - El ancho de la mantisa es de 25 bits.
   - La mantisa está normalizada y la coma se sitúa a la izquierda del bit más significativo.
   - El exponente viene representado en signo-magnitud
   - El ancho del exponente son 8 bits.

Se pide:

a) Calcular el rango de representación para los números enteros.

b) Calcular el rango de representación para los números en coma flotante en el caso de utilizar una mantisa con bit implícito y sin bit implícito.

c) Representar en dicho formato los números:

- 240
- -240
- 24,25 considerando que la mantisa se representa sin bit implícito.
- 24,25 considerando que la mantisa se representa con bit implícito.
- -24,25 considerando que la mantisa se representa sin bit implícito.
- -24,25 considerando que la mantisa se representa con bit implícito.

4. Se tiene un ordenador con los siguientes formatos de representación:
   - Números enteros con 16 bits, representados en complemento a 1.
   - Números en coma flotante con las siguientes características:
   - Mantisa fraccionaria expresada en complemento a 1.
   - El ancho de la mantisa es de 16 bits.
   - La mantisa está normalizada y la coma se sitúa a la izquierda del bit más significativo.
   - El exponente viene representado en exceso 2^{6-1}.
   - El ancho del exponente son 6 bits.

Se pide:

a) Calcular el rango de representación para los números enteros.

b) Calcular el rango de representación para los números en coma flotante en el caso de utilizar una mantisa con bit implícito y sin bit implícito.

c) Representar en dicho formato los números:

- 123
- -124
- 321,23 considerando que la mantisa se representa sin bit implícito.
- 321,23 considerando que la mantisa se representa con bit implícito.

---

5. Se tiene un ordenador con los siguientes formatos de representación:
   - Números enteros con 8 bits, representados en exceso.
   - Números en coma flotante con las siguientes características:
   - Mantisa fraccionaria expresada en complemento a 1.
   - El ancho de la mantisa es de 12 bits.

- La mantisa está normalizada y la coma se sitúa a la izquierda del bit más significativo.
- El exponente viene representado en signo-magnitud.
- El ancho del exponente son 4 bits.

Se pide:

a) Calcular el rango de representación para los números enteros.

b) Calcular el rango de representación para los números en coma flotante en el caso de utilizar una mantisa con bit implícito y sin bit implícito.

c) Representar en dicho formato los números:

- 3
- -324
- 25,123 considerando que la mantisa se representa sin bit implícito.
- 25,123 considerando que la mantisa se representa con bit implícito.
- -2,3 considerando que la mantisa se representa sin bit implícito.
- -2,3 considerando que la mantisa se representa con bit implícito

---

6. Se tiene un ordenador con los siguientes formatos de representación:
   - Números enteros con 6 bits, representados en complemento a 1.
   - Números en coma flotante con las siguientes características:
   - Mantisa fraccionaria expresada en complemento a 2.
   - El ancho de la mantisa es de 40 bits.
   - La mantisa está normalizada y la coma se sitúa a la izquierda del bit más significativo.
   - El exponente viene representado en exceso 2^{16-1}.
   - El ancho del exponente son 16 bits.

Se pide:

a) Calcular el rango de representación para los números enteros.

b) Calcular el rango de representación para los números en coma flotante en el caso de utilizar una mantisa con bit implícito y sin bit implícito.

c) Representar en dicho formato los números:

- 24
- -214,25 considerando que la mantisa se representa sin bit implícito.
- -214,25 considerando que la mantisa se representa con bit implícito.

7. Se tiene un ordenador con los siguientes formatos de representación:
   - Números enteros con 16 bits, representados en signo-magnitud.
   - Números en coma flotante con las siguientes características:
   - Mantisa entera expresada en signo-magnitud
   - El ancho de la mantisa es de 24 bits.
   - El exponente viene representado en exceso 2^{8-1}.
   - El ancho del exponente son 8 bits.

Se pide:

a) Calcular el rango de representación para los números enteros.

b) Calcular el rango de representación para los números en coma flotante.

c) Representar en dicho formato los números:

- 24
- -24
- 24,25
- -24,25

8. Se tiene un ordenador con los siguientes formatos de representación:
   - Números enteros con 16 bits, representados en signo-magnitud.
   - Números en coma flotante con las siguientes características:
   - Mantisa entera expresada en signo-magnitud
   - El ancho de la mantisa es de 32 bits.
   - El exponente viene representado en exceso 2^{8-1}.
   - El ancho del exponente son 8 bits.

Se pide:

a) Calcular el rango de representación para los números enteros.

b) Calcular el rango de representación para los números en coma flotante.

c) Representar en dicho formato los números:
   - 124
   - -214
   - 214,25
   - -124,25

9. Se tiene un ordenador con los siguientes formatos de representación:
   - Números enteros con 6 bits, representados en exceso.
   - Números en coma flotante con las siguientes características:
   - Mantisa fraccionaria expresada en complemento a 2.
   - El ancho de la mantisa es de 14 bits.
   - La mantisa está normalizada y la coma se sitúa a la izquierda del bit más significativo.
   - El exponente viene representado en signo-magnitud.
   - El ancho del exponente son 6 bits.

Se pide:

a) Calcular el rango de representación para los números enteros.

b) Calcular el rango de representación para los números en coma flotante en el caso de utilizar una mantisa con bit implícito y sin bit implícito.

c) Representar en dicho formato los números:

- 3
- -4
- 4,725 considerando que la mantisa se representa sin bit implícito.
- 4,725 considerando que la mantisa se representa con bit implícito.
- -2,725 considerando que la mantisa se representa sin bit implícito.
- -2,725 considerando que la mantisa se representa con bit implícito.

10. Se tiene un ordenador con los siguientes formatos de representación:
    - Números enteros con 16 bits, representados en complemento a 1.
    - Números en coma flotante con las siguientes características:
    - Mantisa fraccionaria expresada en complemento a 2.
    - El ancho de la mantisa es de 12 bits.
    - La mantisa está normalizada y la coma se sitúa a la izquierda del bit más significativo.
    - El exponente viene representado en exceso 2^{6-1}.
    - El ancho del exponente son 6 bits.

Se pide:

a) Calcular el rango de representación para los números enteros.

b) Calcular el rango de representación para los números en coma flotante en el caso de utilizar una mantisa con bit implícito y sin bit implícito.

c) Representar en dicho formato los números:

- -2
- 124,725 considerando que la mantisa se representa sin bit implícito.
- 124,725 considerando que la mantisa se representa con bit implícito.

---

11. El estándar IEEE 754 define la representación de números fraccionarios con 32 y con 64 bits, según los siguientes formatos:
    - Formato simple:
        - Mantisa fraccionaria expresada en signo-magnitud.
        - El ancho de la mantisa es de 24 bits.
        - El ancho del exponente es de 8 bits y se representa en exceso 2^{8-1}-1 (exceso 127).
    - Formato doble:
        - Mantisa fraccionaria expresada en signo-magnitud.
        - El ancho de la mantisa es de 53 bits.
        - El ancho del exponente es de 11 bits y se representa en exceso 2^{11-1}-1 (exceso 1023).

En este formato no todos los patrones de bits se interpretan de la manera habitual, sino que dependiendo de la información almacenada en la mantisa y en el exponente:

- Los números normalizados se representan utilizando los exponentes desde 0...01 hasta 1...10, considerando que la mantisa está

normalizada y utiliza bit implícito, situándose la coma a la derecha de dicho bit.

- Los números no normalizados se representan con exponente 0...0 y la magnitud de la mantisa será distinta de 0...0. (Esta representación se utiliza para representar números cuyo exponente sea muy pequeño, pudiendo tratar de esta manera situaciones de desbordamiento de exponente).
- La representación del ±0 se realiza considerando 0...0 en el exponente y en la magnitud de la mantisa.
- La representación de se realiza considerando el exponente a 1...1 y la magnitud de la mantisa 0...0. (Esta representación es útil para tratar los desbordamientos, como error o considerando el valor ±∞, dependiendo del procesamiento que realice el programa que se esté ejecutando).
- El número representado con exponente 1...1 y con mantisa distinta de 0...0 se emplea para señalar condiciones de excepción.

Se pide:

a) Indicar el rango de representación de los números normalizados.

b) Indicar el rango de representación de los números no normalizados.

c) Representar en el formato IEEE 754 en simple precisión los siguientes números:

- +0 considerando el formato simple.
- +0 considerando el formato doble.
- -0 considerando el formato simple.
- -0 considerando el formato doble.
- +∞ considerando el formato simple.
- +∞ considerando el formato doble.

- -∞ considerando el formato simple.
- -∞ considerando el formato doble.
- 5 considerando el formato simple.
- 5 considerando el formato doble.
- 124,725 considerando el formato simple.
- 124,725 considerando el formato doble.
- -32.100.087,3 considerando el formato simple.
- -32.100.087,3 considerando el formato doble.

12. Sabiendo que el dato recibido por un MODEM y protegido mediante código Hamming es el siguiente:

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

Se pide:

a) Calcular si el número recibido es correcto.

b) Si no es correcto, corregir el número.

13. Se tiene un computador conectado a una red telefónica por la que llegan números en coma flotante relativos a las temperaturas en una cámara frigorífica de carnes. Los datos vienen protegidos mediante código Hamming. El formato de representación es en complemento a 2 tanto para el exponente como para la mantisa que además emplea la técnica del bit implícito.

A nuestro terminal ha llegado el dato siguiente:

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

Se pide:

a) Sabiendo que tanto la mantisa como el exponente se expresan en complemento a 2; que el ancho del exponente es de un bit más que los bits de paridad necesarios para proteger el dato; que el resto de los bits del dato pertenecen a la mantisa y que ésta emplea la técnica del bit implícito se pide calcular el rango de numeración para la representación anterior.

b) Verificar si el número llega correctamente y si fuese necesario realizar la corrección correspondiente.

c) Calcular el valor del número recibido.