I formati Floating Point e conversione di numeri frazionari in binario

Esercizio 1

Convertire il numero 12.125 in binario.

Parte intera

La conversione è quella solita.

```
12 0 12:2=6 (resto 0)
 6 0  6:2=3 (resto 0)
 3 1  3:2=1 (resto 1)
 1 1  1:2=0 (resto 1)
 0
```

12 (base 10) --> **1100** (base 2)

Parte decimale

Si procede con moltiplicazioni successive.

```
0.125 * 2 = 0.250 (riporto 0)
0.250 * 2 = 0.500 (riporto 0)
0.500 * 2 = 1.000 (riporto 1) --> si considera solo
più la parte decimale
0.000 * 2 = FINE
```

Ordinando i riporti al contrario, si ottiene il risultato.

0.125 (base 10) --> **0.001** (base 2)

Il risultato sarà: 12.125 (base 10) --> **1100. 001** (base 2)

Esercizio 2

Convertire il numero 17.55 in binario (precisione 8 cifre decimali, arrotondamento per troncamento).

Parte intera: si usano i metodi già visti. 17 (base 10) --> 10001 (base 2)

Parte decimale: si usano le moltiplicazioni successive.

```
0.55 * 2 = 1.10 (riporto 1) --> si considera solo più "0.10"
0.10 * 2 = 0.20 (riporto 0)
```

```
0.20 * 2 = 0.40 (riporto 0)
0.40 * 2 = 0.80 (riporto 0)
0.80 * 2 = 1.60 (riporto 1) --> si considera solo più "0.60"
0.60 * 2 = 1.20 (riporto 1) --> si considera solo più "0.20"
0.20 * 2 = 0.40 (riporto 0)
0.40 * 2 = 0.80 (riporto 0)
--> si sono ormai calcolate 8 cifre --> si tronca il risultato
```

Il risultato, a 8 cifre binarie frazionali, è quindi pari a : **10001.10001100** (base 2) .

Esercizio 3

Convertire il numero decimale 17.55 con una precisione di 1/100 (decimale)

Il numero è già stato convertito nell'esercizio precedente. Rimane questa volta da capire quante sono le cifre frazionarie che è necessario ricavare.

Si ricorda la seguente proprieta': la precisione di un numero binaario è data dal valore 1/(2^n), dove n è il numero di cifre frazionarie.

Una precisione di 1/100 decimale si ottiene quindi con 1/128, ossia 1/(2^7).

Il numero di bit frazionari necessari è quindi 7.

Il risultato, a 7 cifre binarie frazionali, è quindi pari a : **10001.1000110** (base 2).

Esercizio 4

Convertire i seguenti numeri in binario, con una precisione di 8 cifre decimali.

Decimale	**Binario**
23.466	*10111.01110111*
61.625	*111101.10100000*
13.543	*1101.10001011*
55.110	*110111.00011100*
19.999	*10011.11111111*
22.001	*10110.00000000*
41.700	*101001.10110011*

Esercizio 5

Convertire il numero binario 1101.00101100 in decimale.

Il procedimento è analogo alla conversione della sola parte intera.

Parte intera: `1*2^0 + 0*2^1 + 1*2^2 + 1*2^3 = 1 + 0 + 4 + 8 =` **`13`**

Parte decimale: `0*2^-1 + 0*2^-2 + 1*2^-3 + 0*2^-4 + 1*2^-5 + 1*2^-6 + 0*2^-7 + 0*2^-8 = 0 + 0 + 0.125 + 0 + 0.03125 + 0.015625 + 0 + 0 =` **`0,171875`**

--> 1101.00101100 (Base 2) = **13.171875** (base 10)

Esercizio 6

Convertire i seguenti numeri frazionari binari in decimale.

Binario	**Decimale**
11101.11010111	*29.83984375*
10001.00001101	*17.05078125*
00011.10011001	*3.59765625*
11000.11111001	*24.97265625*
10011.00100100	*19.140625*
10010.01001001	*18.28515625*

Esercizio 7

Convertire il numero decimale 11.876 in rappresentazione reale Floating-Point (virgola mobile) su 32 bit. Indicare, in particolare, come vengono rappresentati mantissa ed esponente.

Mantissa

La trasformazione prevede la conversione in binario del numero 11.876.

Parte intera

La parte intera di X è 11 (base 10) --> **1011** (base 2).

Parte frazionaria

Essendo la parte intera composta da 4 cifre, si dovranno calcolare 20 cifre decimali (la mantissa è composta da 24 cifre binarie).

0.876 (base 10) --> 0.11100000010000011000 (base 2, 20 cifre con approssimazione per troncamento)

La mantissa sarà pari a: 1011.11100000010000011000

Normalizzando, si avrà: 1.01111100000010000011000 * 2^3

Siccome il numero è positivo, e la prima cifra della mantissa (essendo sempre pari a "1") si omette, la mantissa sarà pari a **01111100000010000011000.**

Esponente

Avendo dovuto fare uno shift della mantissa di tre posizioni (si è moltiplicato per 2^3), l'esponente varrà 3.

Essendo però espresso in codice ECCESSO 127, il numero da inserire come esponente sarà 127+3=130.

Si converte quindi questo numero in binario puro con i metodi già visti, ottenendo 10000010.

Il risultato finale sarà:

Segno	Esponente	Mantissa
0	10000010	01111100000010000011000

Esercizio 8

Convertire i seguenti numeri decimali frazionari in codifica standard IEEE 754 SP.

Decimale	**Segno**	**Esponente**	**Mantissa**
-12.72	1	*10000010*	*10010111000010100011101*
+14.375	0	*10000010*	*11001100000000000000000*
-46.188	1	*10000100*	*01110001100000010000011*
+7.99	0	*10000001*	*11111111010111000010100*
-12.56	1	*10000010*	*10010001111010111000010*
+5.54	0	*10000001*	*01100010100011110101110*
-2.21	1	*10000000*	*00011010111000010100011*
+0.07	0	*01111011*	*00011110101110000101000*