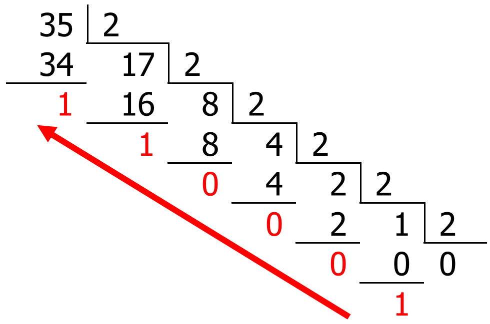
CONVERSIONE DA BASE 10 AD ALTRE BASI E VICEVERSA

I numeri decimali (base 10) li conosciamo tutti, sono i numeri da 0 a 9. Il sistema decimale è posizionale nel senso che lo stesso numero posto in posizioni diverse assume "peso", ossia valore, diverso.

Vi sono altri sistemi di numerazione precisamente il sistema di numerazione **binario (base 2),** in cui i simboli sono {0,1} chiamati bit (**b**inary dig**it**), il sistema **ottale (base 8),** formato dai simboli {0,1,2,3,4,5,6,7}, il sistema **esadecimale (base 16)** i cui simboli sono {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}.

Conversione da base 10 a base 2 e viceversa.

Per convertire un numero da base 10 a base 2 si devono effettuare un insieme di divisioni per 2, conservando tutti i resti, partendo dal numero dato (che si vuole convertire) e continuando a dividere i quozienti che si ottengono ad ogni divisione fino a che l'ultimo quoziente diventa 0. I resti delle divisioni, presi dall'ultimo risalendo fino al primo, ci daranno la conversione in binario del numero decimale dato.
Come esempio vogliamo convertire il numero 35_{10} in binario.



in rosso sono riportati i resti delle varie divisioni.

In definitiva il numero 35_{10} = 100011_2.

Per l'operazione inversa (binario -> decimale) poniamo sotto ogni bit del numero binario una potenza del 2, partendo da 2^0, posto sotto la cifra più a destra del numero, incrementando l'esponente di uno fino a completare tutti i numeri binari:

1	0	0	0	1	1	**f.1**
2^5	2^4	2^3	2^2	2^1	2^0	

sommiamo, quindi, tutte le potenze del 2 in corrispondenza degli 1. Nel nostro esempio:

$$2^5+2^1+2^0 =32+2+1=35_{10}.$$

L'operazione corretta, in realtà, avrebbe dovuto moltiplicare il bit binario per la potenza sottostante corrispondente, sommando tale prodotto al successivo, ossia:

$$1*2^5 + 0*2^4 + 0*2^3 + 0*2^2 + 1*2^1 + 1*2^0 = 2^5+ 0 + 0 + 0 + 2^1+2^0 =32+2+1=35_{10}.$$

Conversione da base 10 a base 8 e viceversa.

Le considerazioni fatte per la conversione da base 10 a base 2 valgono anche per questa conversione sostituendo, nella divisione, il divisore 2 con il divisore 8 e, per la conversione da base 8 a base 10, le potenze del 2 con le potenze dell'8.
Ad esempio convertiamo 35_{10} in base 8:

$$\begin{array}{r|l}
35 & 8 \\ \hline
32 & \begin{array}{r|l} 4 & 8 \\ \hline 0 & 0 \\ \hline 4 & \end{array} \\ \hline
3 &
\end{array}$$

in definitiva: $35_{10} = 43_8$.

Esiste un altro metodo che facilita la conversione in questione e fa uso della conversione binaria vista prima. Questo procedimento consiste di tre passi.
Se vogliamo convertire un numero in base 10 a base 8:
- prima lo convertiamo in base 2,
- raccogliamo le cifre binarie a tre a tre partendo dalla posizione più a destra (cifra meno significativa) del numero binario ottenuto dalla conversione decimale->binaria,
- convertiamo ognuna di queste terzine in decimale; il numero ottenuto è la conversione del numero decimale in ottale.

Se volessimo convertire 35_{10} nel corrispondente numero in base 8 tramite il metodo sopra descritto:

$35_{10} = 100011_2$ -> (dividiamo in terzine partendo da dx) -> 100_2 011_2 -> convertiamo ogni singola terzina in decimale -> $100_2 = 4$ e $011_2 = 3$: il numero 43_8, come sappiamo, è la codifica ottale del numero 35_{10}.

Per convertire da ottale a decimale utilizziamo le potenze dell'8 come nell'esempio del caso binario:

$$\begin{array}{cc} 4 & 3 \\ 8^1 & 8^0 \end{array}$$

il numero decimale corrispondente è quindi: $4*8^1 + 3*8^0 = 4*8 + 3*1 = 32 + 3 = 35_{10}$

Avremmo, però, potuto convertire il numero ottale in decimale attraverso la conversione binaria, convertendo, cioè, ogni cifra ottale in terzine binarie, unire queste terzine in un'unica cifra binaria, convertire questa cifra in decimale:

$4 = 100_2$ -> $3 = 011_2$ -> raggruppiamo le due terzine in un unico numero binario e convertiamolo come abbiamo fatto prima (**f.1**) -> $100011_2 = 35_{10}$.

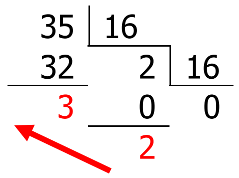
Conversione da base 10 a base 16 e viceversa.

Anche in questo caso le considerazioni fatte per la conversione da base 10 a base 2 valgono per questa conversione sostituendo, nella divisione, il divisore 2 con il divisore 16 e, per la conversione da base 16 a base 10, le potenze del 2 con le potenze di 16.
Siccome, però, il sistema è formato da 16 simboli, comprese delle lettere, diamo di seguito la tabella di codifica esadecimale (HEX), decimale (DEC), binaria (BIN).

HEX	DEC	BIN
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

(tab. 1)

Supponiamo di voler convertire il numero 35_{10} in esadecimale:



quindi $35_{10} = 23_{16}$.

Anche in questo caso esiste un altro metodo che facilita la conversione in questione e che fa uso della conversione binaria vista prima. Questo procedimento consiste di tre passi.
Se vogliamo convertire un numero in base 10 a base 16:
- prima lo convertiamo in base 2,
- raccogliamo le cifre binarie a quattro a quattro partendo dalla posizione più a destra (cifra meno significativa) del numero binario ottenuto dalla conversione decimale->binaria,
- convertiamo ognuna di queste quartine in esadecimale secondo la tabella sopra riportata (tab. 1); il numero ottenuto è la conversione esadecimale del numero decimale.

Quindi per convertire 35_{10} in base 16:

$35_{10} = 100011_2$ -> (dividiamo in quartine partendo da dx) -> 0010_2 0011_2 (notate che abbiamo aggiunto, per uniformità, due 0 all'inizio della prima quartina) -> convertiamo ogni singola quartina in esadecimale -> $0010_2 = 2_{16}$ $0011_2 = 3_{16}$: il numero 23_{16} è la codifica esadecimale del numero 35_{10}.

Per convertire da ottale a decimale, utilizziamo le potenze di 16 come nell'esempio del caso binario (ed ottale):

2	3
16^1	16^0

il numero decimale corrispondente è quindi:

$$2*16^1 + 3*16^0 = 2*16 + 3*1 = 32 + 3 = 35_{10}$$

Avremmo, però, potuto convertire il numero esadecimale in decimale attraverso la conversione binaria, convertendo, cioè, ogni cifra esadecimale in quartine binarie, unire queste quartine in un'unica cifra binaria, convertire questa cifra in decimale:

$2_{16} = 0010_2$ -> $3_{16} = 0011_2$ -> raggruppiamo le due quartine in un unico numero binario e convertiamolo come abbiamo fatto prima (**f.1**) -> $00100011_2 = 35_{10}$.

Perché abbiamo bisogno di tre bit per la rappresentazione binaria di un numero ottale e quattro per un numero esadecimale?

Altri esempi di conversione da decimale a ottale ed esadecimale e viceversa.

Consideriamo il numero decimale 125.

$$125_{10} = 1111101_2.$$

Per la conversione ottale abbiamo:

1111101_2 => dividiamo il numero in terzine => 001_2 111_2 101_2 => 1 7 5

in definitiva $125_{10} = 175_8$.

Per la conversione esadecimale:

1111101_2 => dividiamo il numero in quartine => 0111_2 1101_2 => $0111_2 = 7$ $1101_2 = D$ (vedi tab.1)

in definitiva $125_{10} = 7D_{16}$.

Per la conversione da ottale a decimale:

175_8.=> convertiamo ogni singola cifra ottale in terzine binarie 1 = 001_2 7 = 111_2 5 = 101_2;

raggruppiamo le terzine in un'unica cifra binaria convertendola in decimale:

$$001111101_2 = 125_{10}.$$

Per la conversione da esadecimale a decimale:

$7D_{16}$. => convertiamo ogni singola cifra esadecimale in quartine binarie (possiamo aiutarci con la tab. 1) => 7 = 0111_2 D = 1101_2 => raggruppiamo le quartine in un'unica cifra binaria convertendola in decimale:

$$01111101_2 = 125_{10}.$$