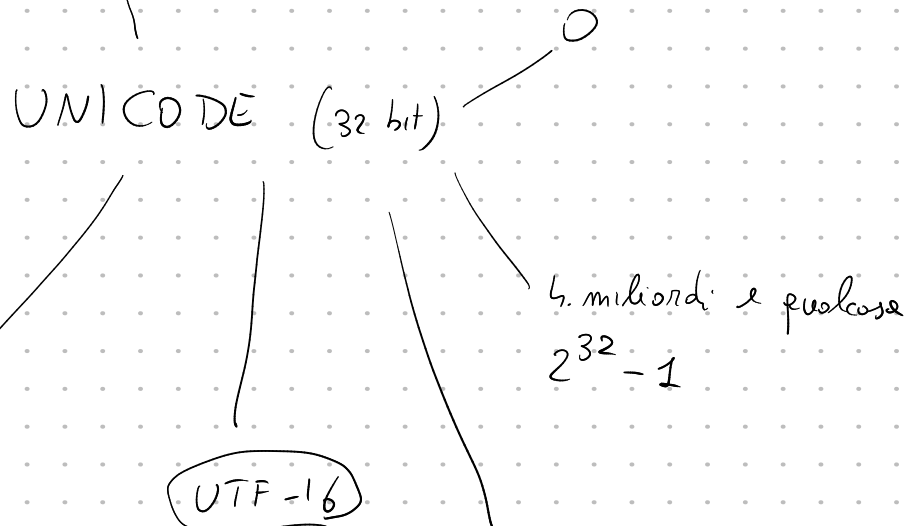
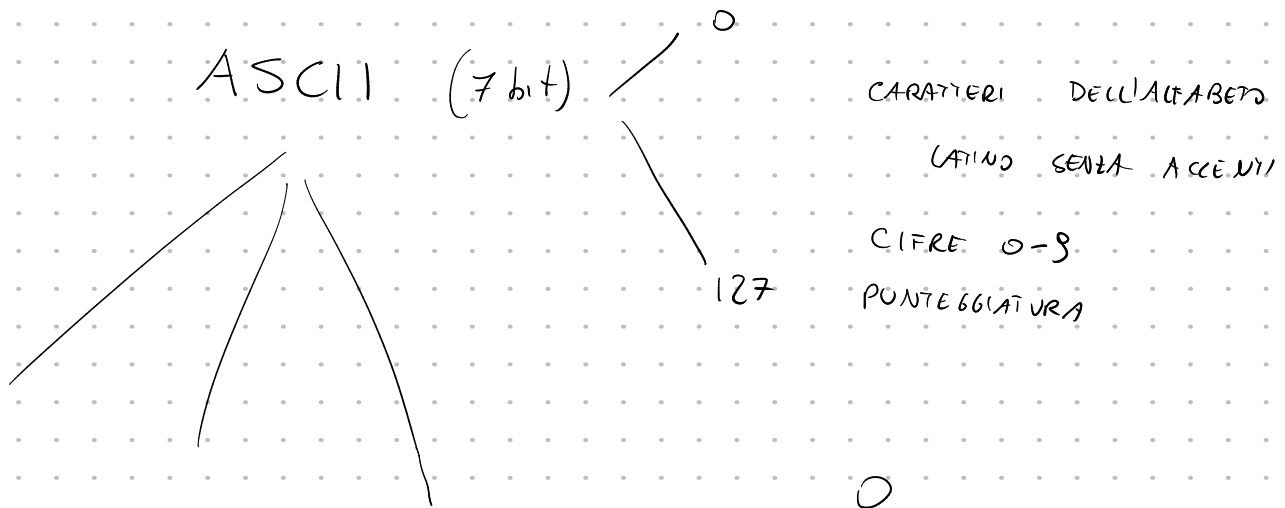


RAPPRESENTAZIONE BINARIA DI CARATTERI



use 1 byte per i codici da 0 e 127, più byte per i codici successivi (compatibile ASCII)

use il codice numerico così com'è e occupa sempre 32 bit

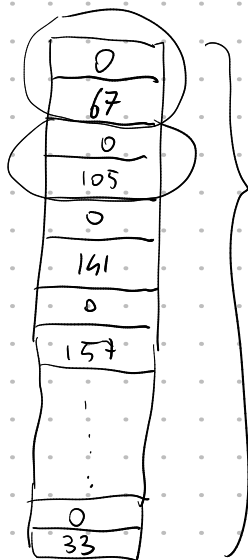
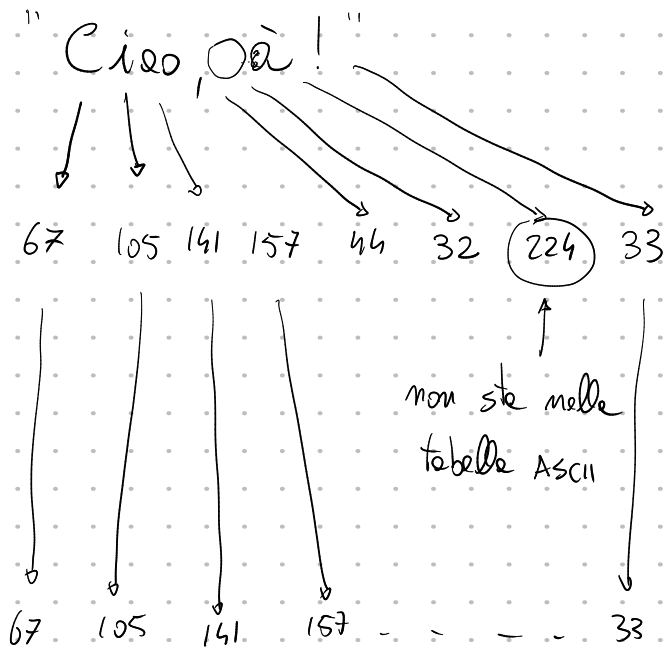
use 2 byte per quasi tutto, 4 byte per i caratteri più strani

Il tipo CHAR di JAVA può codificare solo i caratteri per cui sono sufficienti 2-byte in UTF-16.

STRINGHE

Si mettono in memoria uno dopo l'altro i codici ^{dei caratteri} che compongono la stringa.

Esempio



i byte in memoria che contengono
la rappresentazione della stringa di cui
sopra.

CODICI DI CONTROLLO PER ANDATA A CAPO

LINUX
UNIX
MAC OS_x } LF (10)

Mac OS prima di MAC OS_x } CR (13)

WINDOWS } CR LF (13 10)

CODICE OTTALE ED ESADECIMALE

$453_{10} = 111000101_2$ LUNGI.
DIFFICILI DA RICORDARE

base 8

base 16

$$4 \cdot 8^3 + 0 \cdot 8^2 + 5 \cdot 8^1 = 453$$

$453 : 8 = 56 \text{ (5)}$

$56 : 8 = 7 \text{ (0)}$

$7 : 8 = 0 \text{ (7)}$

705_8

$0 \rightarrow 0$

$1 \rightarrow 1$

...

$8 \rightarrow 8$

$A \rightarrow 10$

$B \rightarrow 11$

$C \rightarrow 12$

$D \rightarrow 13$

$E \rightarrow 14$

$F \rightarrow 15$

$4AB_{16}$

$4 \cdot 16^2 + A \cdot 16^1 + B \cdot 16^0$

$4 \cdot 256 + 10 \cdot 16 + 11 = 1024 + 160 + 11 = 1195_{10}$

$1195 : 16 = 74 \text{ (11)}$

$74 : 16 = 4 \text{ (10)}$

$4 : 16 = 0 \text{ (4)}$

$4 \ 10 \ 11$

$4AB_{16}$

PERCHÉ USARE BASI 16 e 8?

FACILE CONVERTIRE DALLA BASE 2

A QUESTE BASI

$$101101011101_2$$

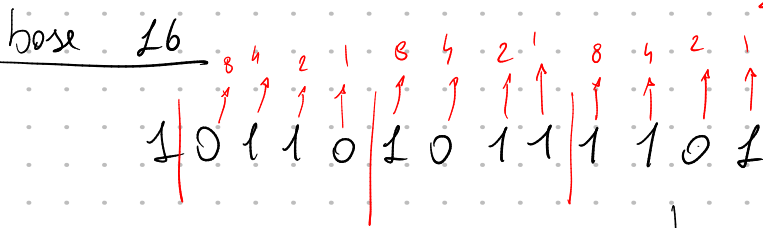
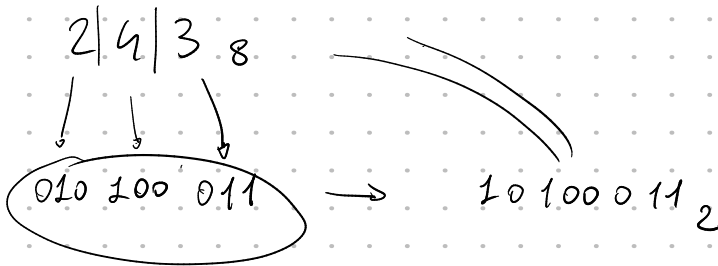
Se voglio convertire da base 5 e base 2

devo passare per la base 10

Per le basi 8 e 16 c'è invece un procedimento rapido



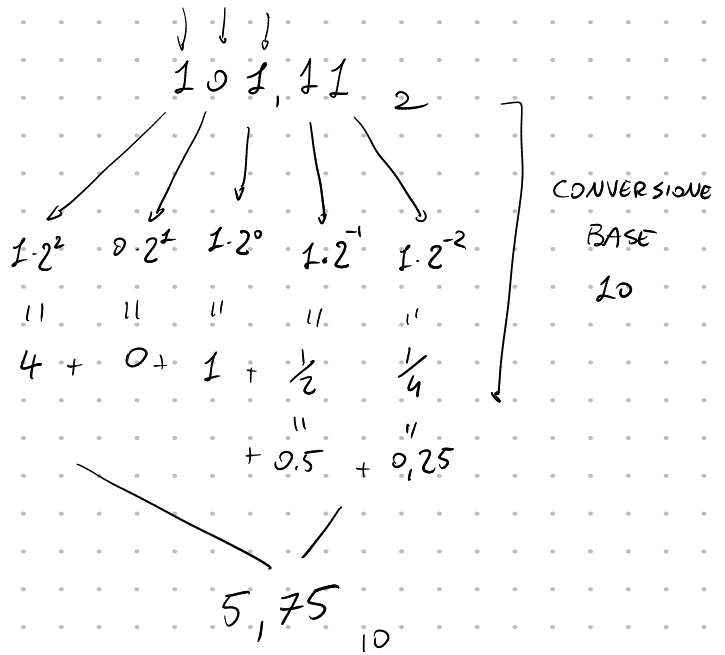
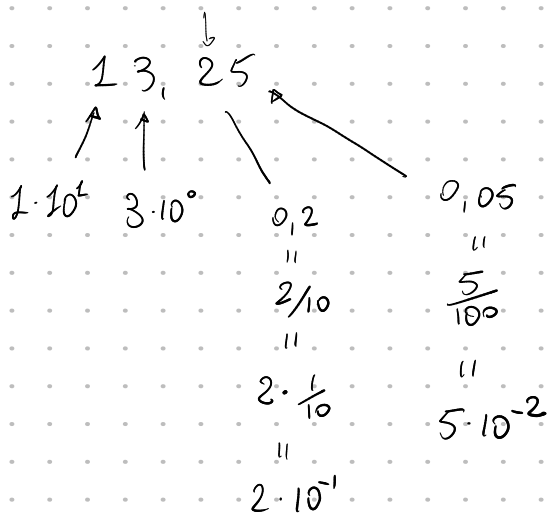
base 8 $1\ 3\ 2\ 7\ 5 \rightsquigarrow 13275_8 =$



base 16 $1\ 6\ 11\ 13 \rightsquigarrow 16B.D_{16}$

5821_{10}

NUMERI CON LA VIRGOLA



13,625 → trovare in base 2?

si esegue le procedure che abbiamo imparato dalle divisioni successive

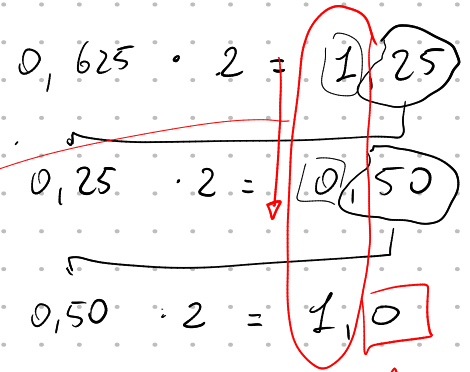
si usa un nuovo algoritmo, basato su MOLTIPLICAZIONI SUCCESSIVE.

$$13_{10} = \underline{1101}_2$$

prendo le parti intere che ho ottenuto dall'alto verso il basso: ottengo il numero in base 2

$$0,625_{10} = \underline{0,101}_2$$

$$\oplus \begin{matrix} 1101 \\ 0,101 \end{matrix}$$



parte dopo la virgola = 0. MI FERMO

CONVERSIONE DI 0,1 in BINARIO

$$0,1_{10} \cdot 2 = 0,2$$

$$0,2 \cdot 2 = 0,4$$

$$0,4 \cdot 2 = 0,8$$

$$0,8 \cdot 2 = 1,6$$

$$0,6 \cdot 2 = 1,2$$

$$0,2 \cdot 2 = 0,4$$

$$0,1_{10} = 0,0001100110011...$$

$$= 0,00011_2$$

$$0,0001100110$$

troncato e lo si apre dopo
la virgola