# Numeri Razionali

32 bit


1 bit 8 bit 23 bit

Float

$$X = (\pm)\, m \times 2^{e}$$

$m = 1.\text{---}$

bit mantisse: $(1 \times 2^{-1}, 2^{-2}, \text{---}, 2^{-23}.$

8 bit espon: 255 interi: $-127, \text{--}, 0, \text{--} 128.$

$-126, \text{--}, 0, \text{--} 127$

$e: e^{-126}$ $\quad m = 2^{-23}$

num. piccoli: $2^{-149}$

Double Precision / Doppia Precisione Double

64 bit totale


1 11 bit 52 bit

segno

32 bit

| Classe dei numeri | Minimo | Massimo |
|---|---|---|
| interi | | |
| senza segno | 0 | 4 294 967 295 |
| con segno | −2 147 483 647 | 2 147 483 647 |
| in complemento a 2 | −2 147 483 648 | 2 147 483 647 |
| razionali (in modulo) | | |
| in singola precisione | $1.401\,298 \times 10^{-45}$ | $3.402\,823 \times 10^{38}$ |
| in doppia precisione | $4.940\,656 \times 10^{-324}$ | $1.797\,693 \times 10^{308}$ |

32 bit

64 bit

$\emptyset$: esp = 00000000 , m = 00000000 tutti bit null.

$+\infty$: S=0 esp = 11111111 , m = 000 --- 0

$-\infty$: S=1 esp = 11111111 , m = 0 --- 0

$\sqrt{-1}$ = NaN = Not a Number

esp = 11111111 m = un solo bit $\neq \emptyset$.

## Problema di approssimazioni

Base 10 e 3 cifre decimali (equivalente a 3 bit)

$a = 1000 = 1.000 \times 10^3$

$b = 999.8 = 9.998 \times 10^2$

$= 0.9998 \times 10^3$

$a - b = 1000 - 999.8 = 0.2$

Calcolo in virgola mobile

$a - b = (1.000 - 0.999) \times 10^3 = 0.001 \times 10^3 = 1$

$a - b = 1$ con 3 cifre

Con 4 cifre decimali:

$a - b = 1.0000 \times 10^3 - 0.9998 \times 10^3$

$= 0.0002 \times 10^3 = 0.2$

## Rapp Caratteri

Mappe tra caratteri e numeri interi.

Anni 60 codice ASCII a 8 bit di informazione

# ASCII TABLE

| Decimal | Hex | Char | Decimal | Hex | Char | Decimal | Hex | Char | Decimal | Hex | Char |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | [NULL] | 32 | 20 | [SPACE] | 64 | 40 | @ | 96 | 60 | ` |
| 1 | 1 | [START OF HEADING] | 33 | 21 | ! | 65 | 41 | A | 97 | 61 | a |
| 2 | 2 | [START OF TEXT] | 34 | 22 | " | 66 | 42 | B | 98 | 62 | b |
| 3 | 3 | [END OF TEXT] | 35 | 23 | # | 67 | 43 | C | 99 | 63 | c |
| 4 | 4 | [END OF TRANSMISSION] | 36 | 24 | $ | 68 | 44 | D | 100 | 64 | d |
| 5 | 5 | [ENQUIRY] | 37 | 25 | % | 69 | 45 | E | 101 | 65 | e |
| 6 | 6 | [ACKNOWLEDGE] | 38 | 26 | & | 70 | 46 | F | 102 | 66 | f |
| 7 | 7 | [BELL] | 39 | 27 | ' | 71 | 47 | G | 103 | 67 | g |
| 8 | 8 | [BACKSPACE] | 40 | 28 | ( | 72 | 48 | H | 104 | 68 | h |
| 9 | 9 | [HORIZONTAL TAB] | 41 | 29 | ) | 73 | 49 | I | 105 | 69 | i |
| 10 | A | [LINE FEED] | 42 | 2A | * | 74 | 4A | J | 106 | 6A | j |
| 11 | B | [VERTICAL TAB] | 43 | 2B | + | 75 | 4B | K | 107 | 6B | k |
| 12 | C | [FORM FEED] | 44 | 2C | , | 76 | 4C | L | 108 | 6C | l |
| 13 | D | [CARRIAGE RETURN] | 45 | 2D | - | 77 | 4D | M | 109 | 6D | m |
| 14 | E | [SHIFT OUT] | 46 | 2E | . | 78 | 4E | N | 110 | 6E | n |
| 15 | F | [SHIFT IN] | 47 | 2F | / | 79 | 4F | O | 111 | 6F | o |
| 16 | 10 | [DATA LINK ESCAPE] | 48 | 30 | 0 | 80 | 50 | P | 112 | 70 | p |
| 17 | 11 | [DEVICE CONTROL 1] | 49 | 31 | 1 | 81 | 51 | Q | 113 | 71 | q |
| 18 | 12 | [DEVICE CONTROL 2] | 50 | 32 | 2 | 82 | 52 | R | 114 | 72 | r |
| 19 | 13 | [DEVICE CONTROL 3] | 51 | 33 | 3 | 83 | 53 | S | 115 | 73 | s |
| 20 | 14 | [DEVICE CONTROL 4] | 52 | 34 | 4 | 84 | 54 | T | 116 | 74 | t |
| 21 | 15 | [NEGATIVE ACKNOWLEDGE] | 53 | 35 | 5 | 85 | 55 | U | 117 | 75 | u |
| 22 | 16 | [SYNCHRONOUS IDLE] | 54 | 36 | 6 | 86 | 56 | V | 118 | 76 | v |
| 23 | 17 | [END OF TRANS. BLOCK] | 55 | 37 | 7 | 87 | 57 | W | 119 | 77 | w |
| 24 | 18 | [CANCEL] | 56 | 38 | 8 | 88 | 58 | X | 120 | 78 | x |
| 25 | 19 | [END OF MEDIUM] | 57 | 39 | 9 | 89 | 59 | Y | 121 | 79 | y |
| 26 | 1A | [SUBSTITUTE] | 58 | 3A | : | 90 | 5A | Z | 122 | 7A | z |
| 27 | 1B | [ESCAPE] | 59 | 3B | ; | 91 | 5B | [ | 123 | 7B | { |
| 28 | 1C | [FILE SEPARATOR] | 60 | 3C | < | 92 | 5C | \ | 124 | 7C | \| |
| 29 | 1D | [GROUP SEPARATOR] | 61 | 3D | = | 93 | 5D | ] | 125 | 7D | } |
| 30 | 1E | [RECORD SEPARATOR] | 62 | 3E | > | 94 | 5E | ^ | 126 | 7E | ~ |
| 31 | 1F | [UNIT SEPARATOR] | 63 | 3F | ? | 95 | 5F | _ | 127 | 7F | [DEL] |

$2^8$ Caratteri $= 256$

0, ..., 127: Caratteri noti

128 – 255: Uso personale

UNICODE: $17 \times 2^{16}$ Caratteri: l'ultimo 112 Caratt.

in uso $\sim 96000$

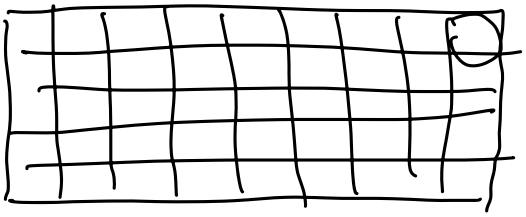
I primi 256 Caratt. di UNICODE $\equiv$ ASCII

A: codice 65 a: codice 97

$65 = 64 + 1$ 01000001

$97 = 65 + 32$ 01100001

## Immagini, Suono



pixel

Colore = RGB

0...255 8 bit 8 bit 8 bit 24 bit #RGB

8 bit

8 bit di luminosità

32 bit per punto per colore + luminosità

Schermo 4K: $\approx$ $4000 \times 1000$ $\sim 4 \times 10^6$ pixel.

# bit $= 4 \times 10^6 \times 32$ bit 4 Byte

$= 16 \times 10^6$ byte $\approx$ 10 MB

Frequenza immagine (refresh rate) 100 Hz $\Rightarrow$ $100 \times 10$ MB/s $= 1000$ MB/s

Hz $= s^{-1}$ $\approx$ 1 GB/sec.