

Numeri razionali

32 b.t

| 1 bit | 8 bit | 23 bit |
|-------|-----------|----------|
| signo | esponente | mantissa |

float

$$x = (\pm) m \times 2^e$$

rapp. normale

$$m = 1.0\ldots$$

e: intero

$$e = a_1 \times 2^{-1} + a_2 \times 2^{-2} + \dots + a_{23} \times 2^{-23}$$

$$m = 1.0010101$$

8 bit esponente: intero \pm rapp. a complemento

$$e \quad 2^8 \text{ numeri} \rightarrow 256 \text{ numeri}$$

$$-127, \dots, 0, 1, \dots, 127, 128$$

$$\text{negativo più piccolo } x = -11111_2$$

$$\text{positivo più piccolo } x = +00000_2$$

doppia precisione: double num. razionale a 64 bit

| 1 b. | 11 b.t | 52 b.t |
|-------|-----------|----------|
| signo | esponente | mantissa |

rapp in virgola mobile

$$0.0002 \qquad \qquad \qquad 202.03$$

non in virgola mobile

$$2 \times 10^{-4} \qquad \qquad \qquad 2.0203 \times 10^2$$

virgola mobile

int 32 b.t rapp. in base 2 a complemento

float 32 b.t

} virgola mobile

double 64 bit

| Classe dei numeri | Minimo | Massimo |
|-----------------------|-----------------------------|----------------------------|
| interi | | |
| senza segno | 0 | 4294967295 |
| con segno | -2147483647 | 2147483647 |
| in complemento a 2 | -2147483648 | 2147483647 |
| razionali (in modulo) | | |
| in singola precisione | 1.401298×10^{-45} | 3.402823×10^{38} |
| in doppia precisione | $4.940656 \times 10^{-324}$ | 1.797693×10^{308} |

RAND-MAX
int
float
double

$$\begin{array}{r} -2^3 \\ 2 \times 2 \\ \hline -127 \end{array}$$

Calcolatori a 32 b.t o 64 b.t?

32 b.t: indirizzo di memoria a 32 b.t

64 b.t: ind. di memoria a 64 b.t

Virgola mobile float

$$\phi : \text{esponente} = 00000000 \quad m = 00000000$$

tutti b.t $\leq \phi$

$$+\infty : s=0 \quad e=11111111 \quad m=00000000000000000000$$

$$x = s \cdot \underbrace{m}_{\textcircled{m}} \cdot 2^e$$

$$-\infty : s=1 \quad e=11111111 \quad m=\emptyset \emptyset - \emptyset$$

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

$$\text{esp} = 11111111 \quad m = \text{un solo b.t} \neq \emptyset$$

nella libreria matematica: M-PI

M-PIZ

Problemi di approssimazione in virgola mobile

Base 10 con 3 cifre (3 b.t decimali)

$$a = 1000 = 1.\boxed{000} \times 10^3$$

$$b = 999.8 = 0.\boxed{999}8 \times 10^3$$

$$a - b = 1000 - 999.8 = 0.2$$

Calcolo in virgola mobile %

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

$a - b = 1$ in virgola mobile

problema si risolve già con 4 bit +

$$a - b = 1.0000 \times 10^3 - 0.9998 \times 10^3 = \\ = 0.0002 \times 10^3 = 0.2$$

problema si risolve nel calcolo a cifre/bit finite

Rapp. di Caratteri:

Mapp 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 | 1 | 125 | 7D | } |
| 30 | 1E | [RECORD SEPARATOR] | 62 | 3E | > | 94 | 5E | ^ | 126 | 7E | ~ |
| 31 | 1F | [UNIT SEPARATOR] | 63 | 3F | ? | 95 | 5F | - | 127 | 7F | [DEL] |

Char

Codice ASCII: $2^8 = 256$ caratteri

0, --, 127: caratteri comuni

128, --, 255: riservati = uso personale

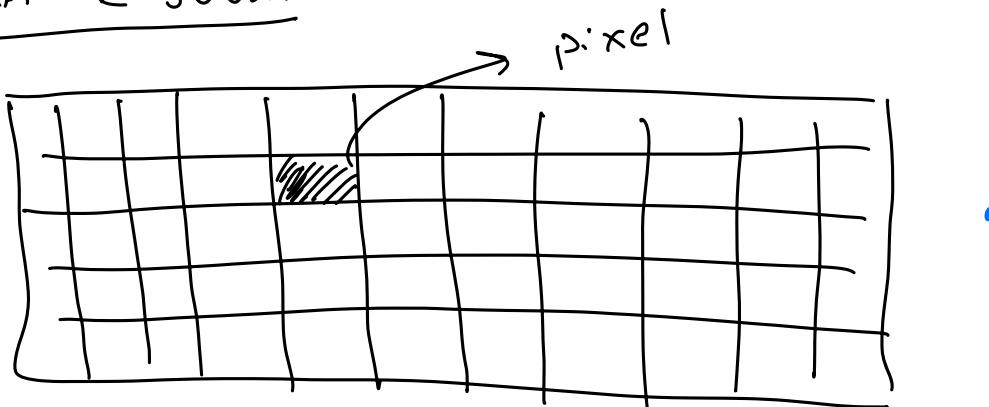
UNICODE: 17×2^{16} caratteri: 114-112 caratteri
attualmente in uso n. 96 000

I primi 256 caratteri UNICODE \equiv ASCII

$$A: \text{Codice GS} = 41_{16} \quad GS = 65 + 1$$

$$65_{10} = 41_{16} = 01000001$$

I Immagini: e suon:



Colore in Codice RGB Red - Green - Blue

$$\begin{aligned} \text{codice} &= R \ G \ B \\ &\quad \left. \begin{array}{c} 0, \dots, 255 \\ 8 \text{ b.t} \end{array} \right\} 0, \dots, 255 \\ &\quad \left. \begin{array}{c} 0, \dots, 255 \\ 8 \text{ b.t} \end{array} \right\} 0, \dots, 255 \\ &\quad \quad \quad = 24 \text{ b.t} \\ &\quad \quad \quad + 8 \text{ b.t} \\ &\quad \quad \quad \text{luminosità} \end{aligned}$$

32 b.t per pixel / punto

$$\text{schermo 4K} \approx 4000 \times 1000 \approx 4 \times 10^6 \text{ pixel}$$

$$\begin{aligned} \text{immagine} \neq \text{b.t} &= 4 \times 10^6 \times 32 \text{ b.t} \approx 16 \times 10^6 \times (8 \text{ b.t}) \\ &\quad \text{Byte} \\ &\approx 10 \text{ MB} \end{aligned}$$

$$\begin{aligned} \text{Freq di refresh} &= 100 \text{ Hz} \Rightarrow 100 \times 10 \text{ MB/sec} = \\ &\quad = 1000 \text{ MB/s} \leq 1 \text{ GB/s} \end{aligned}$$

Calcolatore Elettronico

Memonic contiene dati in digitale

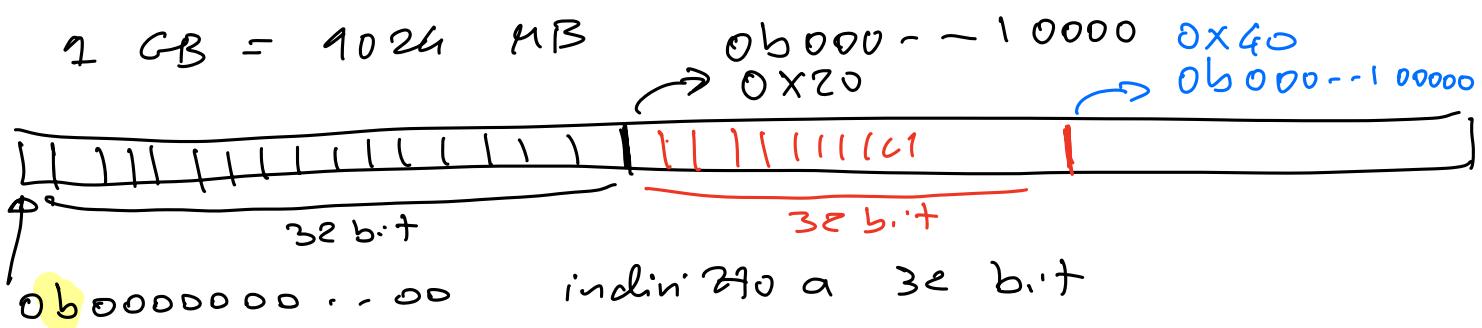
Macchina con 128 GB di memoria

$$128 \times 1024 \times 1024 \times 1024 \times 8 \text{ bit}$$

$$1 \text{ KByte} = 1024 \text{ byte}$$

$$1 \text{ MB} = 1024 \text{ KB}$$

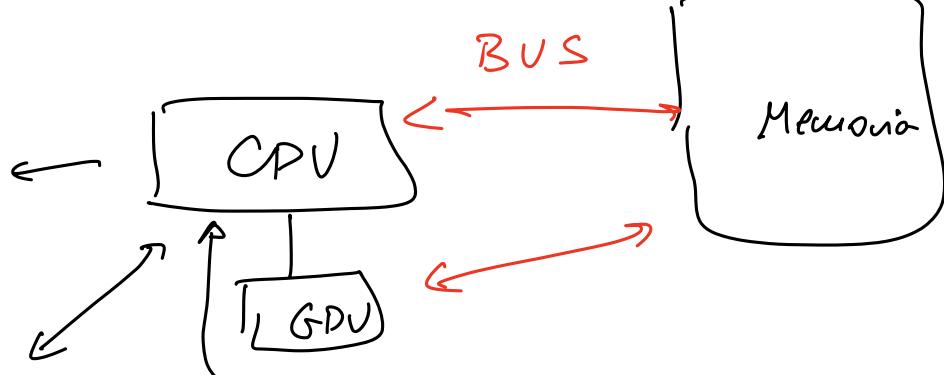
$$1 \text{ GB} = 1024 \text{ MB}$$



| | | | |
|-------|--------------|-------------------|-----------|
| 0b110 | → in base 2 | 110 ₂ | X = 0b110 |
| 110 | → in base 10 | 110 ₁₀ | |
| 0x0 | → in base 16 | 0 ₁₆ | |

Indirizzo di memoria.

Central processing unit



periferiche (accessori)

CLOCK del CPU: velocità di operazioni/sec Hz

GDU: Graphical processing unit

Calcolatore a 8 bit

Operazione di $3 + 2$

operation code

| opCode | Significato | Codice | Parametri |
|-----------|---|--------|-----------|
| 0000 0000 | assegna al registro a il valore contenuto nella locazione di memoria indicata dal parametro che segue | set_a | 1 |
| 0000 0001 | assegna al registro b il valore contenuto nella locazione di memoria indicata dal parametro che segue | set_b | 1 |
| 0000 0010 | somma il contenuto dei registri a e b, quindi mette il risultato nel registro a | sum_ab | 0 |
| 0000 0011 | salta alla locazione indicata dal parametro che segue | goto | 1 |
| 0000 0100 | scrive il contenuto del registro a nella memoria grafica (provocandone la comparsa sullo schermo) | out_a | 0 |
| 0000 0101 | incrementa di 1 il contenuto del registro a | inc_a | 0 |

$a = 3$

Seta 3

[0000 0000] [0000 0011]

Setb 2

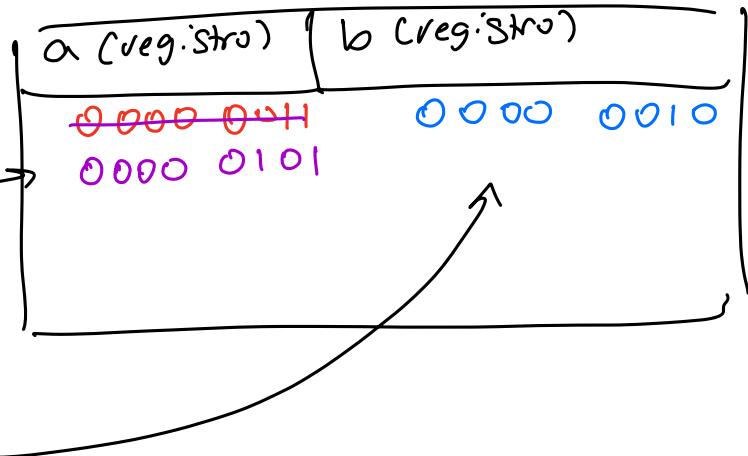
[0000 0001] [0000 0010]

Sumab

[0000 0010]

out a

[0000 0100]



Schermo

