

# Concetti fondamentali

---

September 15, 2018



# Introduzione

---

# Cosa è l'informatica?

E' una scienza, ovvero una conoscenza sistematica di tecniche/metodi per:

- Rappresentare l'informazione
- Elaborare l'informazione tramite algoritmi

## Il concetto di informazione

---

# Cosa è l'informazione?

- Riuscite a leggere questa frase?

L FRS H L VCL MNCNT

- Probabilmente sì, perchè non c'è troppa informazione nelle vocali.
- L'informazione è ciò che, date almeno due occorrenze possibili, supera un'incertezza e risolve un'alternativa.

# Cosa è l'informazione?

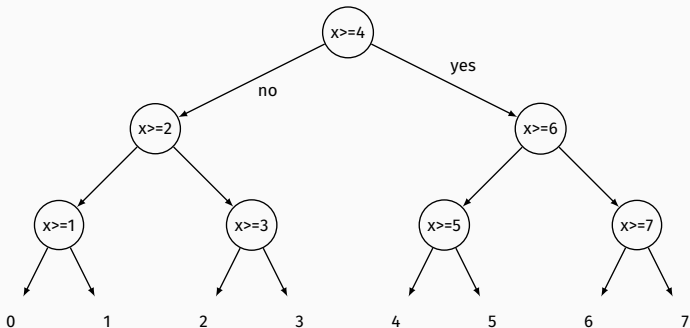
- L'informazione contenuta in un messaggio misura quanto questo ci può sorprendere.
- Un modo per misurare questa cosa è capire quanto sia difficile indovinare l'informazione.

# Cosa è l'informazione?

- Quanta informazione è contenuta in un numero?
- Consideriamo i numeri da 0 a 7. Quante domande servono per indovinarne uno?



# Cosa è l'informazione?



# Definizione formale di informazione

Dati:

- un alfabeto di simboli  $A_N = \{s_1, s_2, \dots, s_i, \dots, s_N\}$
- una canale di comunicazione tra due punti  $X$  e  $Y$  su cui vengono trasmessi tali simboli:  $X \rightarrow Y$

l'informazione trasmessa viene definita come la riduzione di incertezza che si poteva avere a priori sul simbolo trasmesso:

$$I = -\log_2 P_i$$

dove  $P_i$  è la probabilità di trasmissione di quel simbolo.

# Esempio

Dati:

- un alfabeto di simboli  $A_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$
- una canale di comunicazione tra due punti  $X$  e  $Y$  su cui vengono trasmessi tali simboli:  $X \rightarrow Y$

avrò che, considerando i simboli **equiprobabili** l'informazione di ciascun simbolo sarà:

$$I = -\log_2 \frac{1}{8} = 3$$

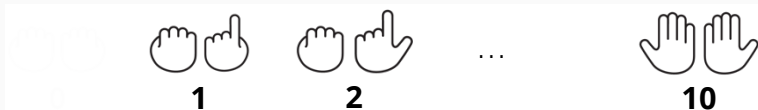
# Informazione numerica

---

Uno dei più frequenti usi del calcolatore è quello di elaborare informazioni numeriche. Per far questo, deve essere in grado di **rappresentare** l'informazione numerica.

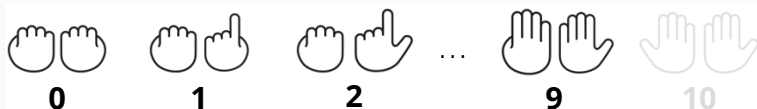
# L'informazione numerica

Il nostro sistema di numerazione è il sistema **decimale**. Ciò ha probabilmente origine dal fatto che abbiamo 10 dita che possiamo usare per poter comunicare un numero.

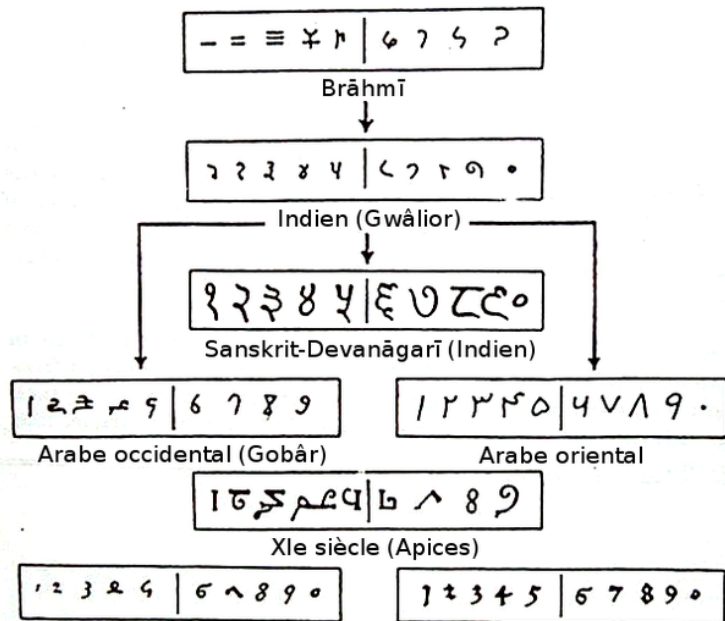


# L'informazione numerica - codifica posizionale

Un'intuizione fondamentale quasi certamente dovuta agli indiani nei primi secoli d.C., è che, per poter rappresentare in maniera compatta un numero maggiore o uguale a 10, è necessario introdurre il concetto di 0 (il **vuoto**) e limitarsi a rappresentare fino al 9.



# L'informazione numerica - codifica posizionale





# L'informazione numerica - codifica posizionale

- Quindi, per comunicare un numero si può utilizzare più di una segnalazione (o simbolo), assegnando un peso differente a ciascuna di essi in base alla loro posizione.
- Ad esempio per comunicare il numero 59, usiamo due segnalazioni:



**5**

Implicitamente  
moltiplicato per 10

.....



**9**

Implicitamente  
moltiplicato per 1

# L'informazione numerica - codifica posizionale

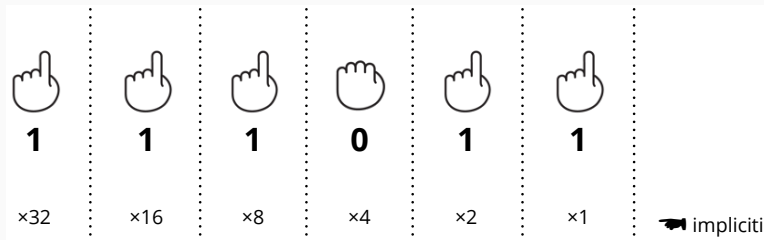
Il sistema decimale è un **sistema posizionale**, ovvero le cifre di un numero rappresentano il **peso** che viene moltiplicato ad una potenza del 10 a seconda della loro **posizione**:

$$59_{10} = 5 \cdot 10^1 + 9 \cdot 10^0$$

- Un computer ragiona sulla base di soli due stati, passa corrente (acceso) o non passa corrente (spento): è come se avesse una sola mano e un solo dito!.
- Per capire come effettivamente il calcolatore rappresenta i numeri, cosa avrebbe fatto la civiltà umana con una sola mano e un solo dito?

# L'informazione numerica - codifica binaria

Avrebbe usato pesi differenti (potenze del due) e più segnalazioni!



# L'informazione numerica - codifica binaria

- Formalmente, l'insieme di simboli che può utilizzare ha solo due elementi:

$$A_2 = \{0, 1\}$$

e viene detto alfabeto **binario**; l'unità di informazione in questo caso è chiamata **bit**.

- Il valore implicito nella rappresentazione è quindi calcolato secondo questa formula.

$$b_5 \cdot 2^5 + b_4 \cdot 2^4 + b_3 \cdot 2^3 + b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

dove ogni bit  $b_i \in A_2 = \{0, 1\}$ .

Soluzione:

$$59_{10} = (111011)_2$$

Poiché:

$$1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 32 + 16 + 8 + 2 + 1 = 59$$

# Elaborazione dell'informazione / Gli algoritmi

---

# Cosa è un algoritmo

E' la modalità con cui descriviamo la soluzione ad un problema. Può essere definito come una sequenza finita di operazioni elementari tali che:

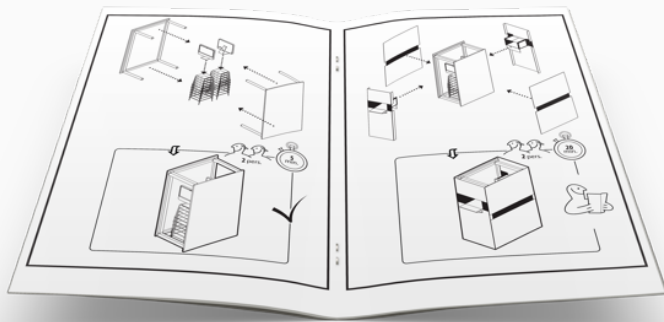
- siano comprensibili ad uno specifico esecutore
- possano essere eseguite senza ambiguità
- permettano di risolvere uno specifico problema



# Firma di un algoritmo

La firma di un algoritmo descrive i dati che l'algoritmo riceve in ingresso (**input**) e quella dei dati in uscita (**output**):

$$\text{nomeAlgoritmo} : (\text{in}_1, \text{in}_2, \dots, \text{in}_n) \rightarrow \text{out} \quad (1)$$



Le istruzioni Ikea sono un algoritmo la cui firma è:

$$\text{montaLibreria} : (\text{pezzo}_1, \text{pezzo}_2, \dots, \text{pezzo}_n) \rightarrow \text{libreria} \quad (2)$$

Un algoritmo è caratterizzato quindi da:

1. una **descrizione** (e.g. grafica, o testuale)
2. una **firma**
3. un **esecutore** cioè chi esegue i passi specificati nella **descrizione** (ad esempio, nel caso precedente l'esecutore è umano).

Esistono altri modi di descrivere gli algoritmi? Sicuramente, ed il primo che vedremo è il **diagramma di flusso**.

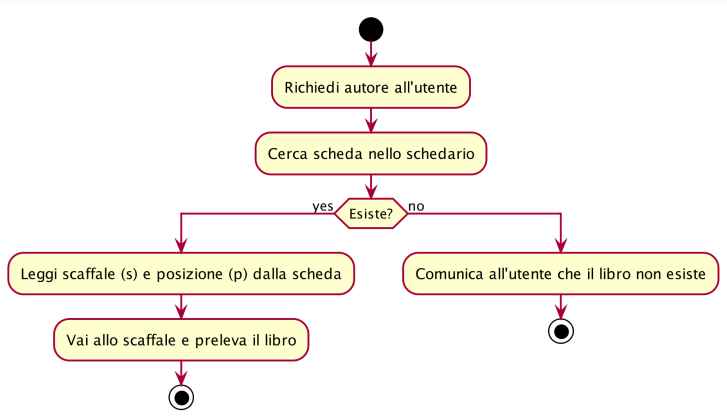
## Esempio: algoritmo di ricerca di un libro in biblioteca

- Firma:

$\text{cercaLibro} : \text{nomeAutore} \rightarrow \text{libro} \quad (3)$

- Esecutore: bibliotecario
- Descrizione: (vedi pagina successiva)

# Esempio: algoritmo di ricerca di un libro in biblioteca



## Esempio: algoritmo di ricerca di un libro in biblioteca

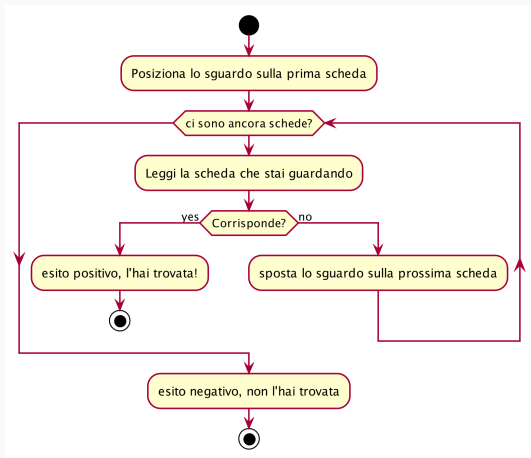
Come è fatto “cerca nello schedario”? Di fatto è un sotto-algoritmo. Possiamo descriverlo utilizzando la stesso tipo di informazioni:

- Firma:

$\text{cercaSchedaInSchedario} : \text{nomeAutore} \rightarrow \text{Scheda} \quad (4)$

- Esecutore: bibliotecario
- Descrizione: (vedi pagina successiva)

# Esempio: algoritmo di ricerca di un libro in biblioteca



è veramente la soluzione migliore? no. Vedremo più avanti che ne esiste un'altra.

# A chi serve un diagramma di flusso?

- I *diagrammi di flusso* permettono di descrivere un algoritmo indipendentemente dal linguaggio di programmazione usato.
- Servono tipicamente a esecutori umani, ma anche a progettisti umani di algoritmi (i **programmatori!**).



## Esempio: Calcolo massimo comun divisore

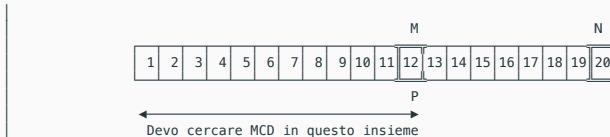
- Supponiamo di voler descrivere l'algoritmo per il calcolo del MCD.
- Innanzitutto definiamo la firma dell'algoritmo:

$\text{calcolaMCD} : (\text{numero } N, \text{numero } M) \rightarrow \text{MCD di } M \text{ ed } N \quad (5)$

- Esecutore: chiunque sappia fare dei calcoli
- Descrizione: (*mumble mumble*)

# Esempio: Calcolo massimo comun divisore

- Per progettarlo, proviamo a crearci un modello mentale, studiando uno o più esempi concreti.
- Ad esempio, come posso derivare il massimo comun divisore tra  $M=12$  ed  $N=20$
- Innanzitutto, esiste un intervallo preciso in cui questo può esistere:



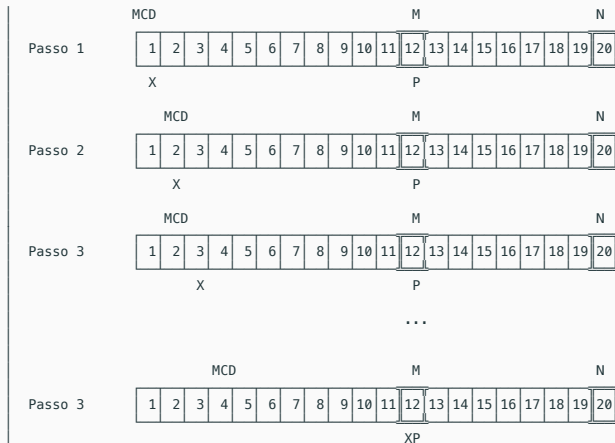
## Esempio: Calcolo massimo comun divisore

- La maniera più semplice è la **ricerca sequenziale**:
  - Mentalmente è come se procedessi sequenzialmente focalizzandomi su ogni numero dell'intervallo alla volta, che segnerò con X.
  - Ad ogni passo, controllo che X divida sia M che N. Se lo fa, lo segnerò con la sigla MCD premurandomi di cancellare MCD dai numeri precedenti.

## Esempio: Calcolo massimo comun divisore

- La maniera più semplice è la **ricerca sequenziale**:
  - Mentalmente è come se procedessi sequenzialmente focalizzandomi su ogni numero dell'intervallo alla volta, che segnerò con X.
  - Ad ogni passo, controllo che X divida sia M che N. Se lo fa, lo segnerò con la sigla MCD premurandomi di cancellare MCD dai numeri precedenti.

# Esempio: Calcolo massimo comun divisore

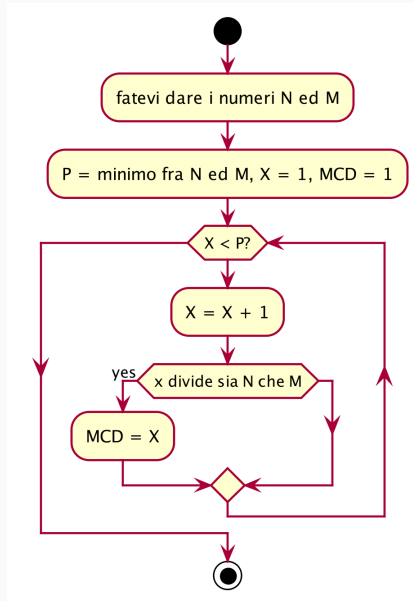


Quando sono arrivato alla fine dell'intervallo ( $X=P$ ), l'ultimo numero segnato con MCD sarà la soluzione del mio problema.

## Esempio: Calcolo massimo comun divisore

- Provate a calcolare il MCD di  $N=4$  e  $M=12$  con questo sistema.
- E' possibile descrivere la procedura per qualsiasi  $M$  ed  $N$  (**generalizzazione**)?

# Esempio: Calcolo massimo comun divisore



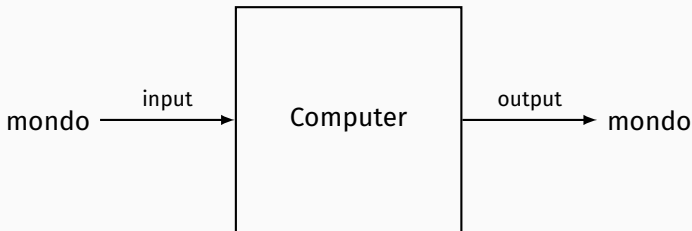
# Elaborazione dell'informazione / Il calcolatore

---



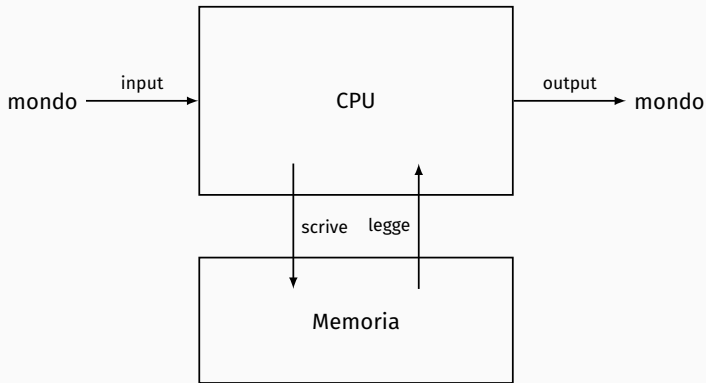
# Cosa è un calcolatore

- Nell'epoca moderna, è uno strumento elettronico che aiuta a *rappresentare* ed *elaborare* l'informazione in ingresso (`input`), producendone una in uscita (`output`) tramite un algoritmo (programma).
- In generale, l'informazione può variare nel tempo.

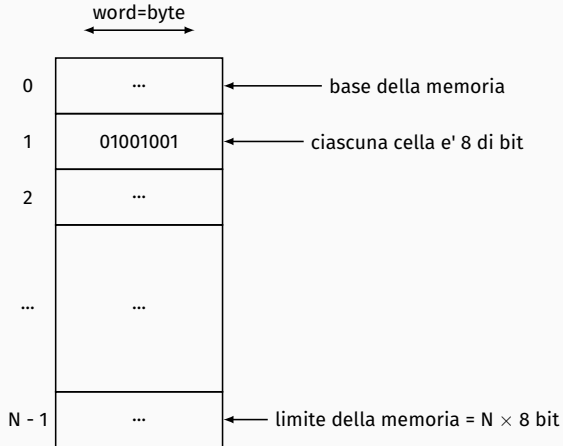


- Il calcolatore (in particolare la Central Processing Unit - CPU) può eseguire velocemente algoritmi composti da operazioni semplici.
- Questi algoritmi sono detti programmi in linguaggio macchina e sono memorizzati, insieme alle variabili, nella **memoria centrale** del sistema.

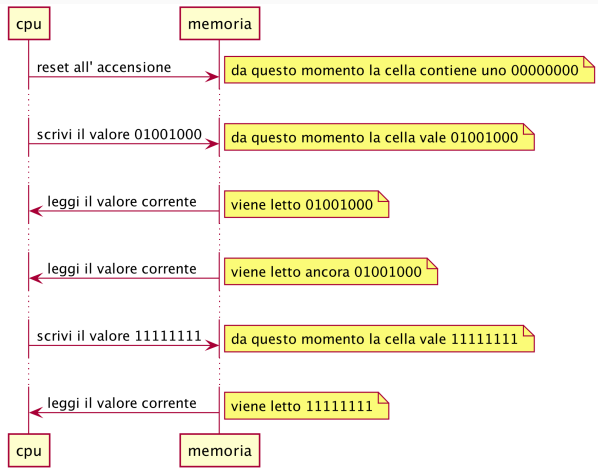
# Cenni di architettura del calcolatore



# Memoria centrale



# Vita di una cella di memoria



- **I programmi.** Il linguaggio macchina o **codice macchina** è il linguaggio in cui sono scritti i programmi eseguibili per computer; è basato su un alfabeto **binario** poiché deve poter essere letto (e scritto) dalla memoria dinamica.
- **I dati.** Anche i valori delle variabili sono mantenuti in memoria, utilizzando la **codifica binaria dell'informazione**.

# Cosa va nella memoria

La memoria viene quindi suddivisa in “zone”, ovvero la cpu stabilisce quali celle devono contenere i programmi, e quali celle devono contenere le variabili.

In maniera molto sintetica possiamo dire che la memoria è divisa in due parti, una per contenere le **istruzioni**, e l'altra per contenere i **dati**.



## Elaborazione avanzata dell'informazione

---



- E' impensabile scrivere programmi di media complessità in linguaggio macchina. Si preferisce utilizzare linguaggi più "umani" e convertirli in linguaggio macchina.
- Tali linguaggi sono definiti da una **sintassi** e una **semantica**.

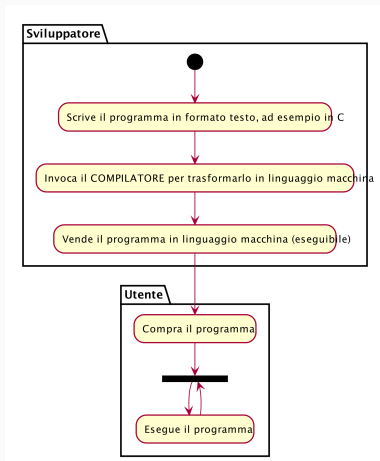
# Linguaggi di alto livello

Descrizione in  
linguaggio C  
dell'algoritmo di  
calcolo del massimo  
comun divisore.  
Il C è il secondo  
linguaggio che  
vedremo in questo  
corso.

```
1  int mcd (int n, int m)
2  {
3      int p, x = 1, mcd = 1;
4      if(n>m) {
5          p = m;
6      } else {
7          p = n;
8      }
9      while(x < p) {
10         x = x + 1;
11         if(!(n % x) && !(m % x)) {
12             mcd = x;
13         }
14     }
15     return mcd;
16 }
```

Come facciamo quindi a creare un programma da un linguaggio di alto livello?

Un approccio è quello di compilare il programma e distribuire l'eseguibile.



# Interprete

Un altro approccio è quello di usare un programma **interprete**.

- Non viene generato nessun programma in linguaggio macchina.
- Il programma di alto livello viene convertito in tempo reale quando l'utente ne richiede l'esecuzione.

