



Progettazione e sviluppo di applicazioni web

Azienda Speciale di Formazione “Scuola Paolo Borsa”
Ing. Masciadri Andrea <masciadri.andrea@gmail.com>

Chi sono

Andrea Masciadri,

PhD in Computer Science POLITECNICO DI MILANO

Ingegnere informatico: consulente e formatore

Mainly on healthcare research:

Politecnico di Milano, Arts et Métiers ParisTech, Fondazione per la ricerca sulla fibrosi cistica, Centro di Ricerca Arte Musica e Spettacolo

Co-fondatore di LYOTECH SRL

Innovative Start-up, Politecnico di Milano Spin-off (www.lyotech.it)





.. chi siete?



Il corso

1. Cenni di informatica e basi di programmazione (Python)
2. Design del software e della base di dati
3. Sviluppo di applicazioni web
4. Ingegneria del software
(documentazione, progettazione, sviluppo, test, manutenzione)



Oggi

Autovalutazione

BREAK

Introduzione all'informatica



Autovalutazione

<https://forms.gle/7xVuawW9zpMtraPo8>





INFORMAZIONE E COMUNICAZIONE

- **Informazione**

Notizia, dato o elemento che consente di avere conoscenza più o meno esatta di fatti, situazioni, modi di essere, diminuendo l'incertezza

- **Messaggio**

Tutto ciò che porta informazione

- **Comunicazione**

Scambio di informazione



COS'È L'INFORMATICA?

- **Esistono varie definizioni:**
 - Scienza degli elaboratori elettronici (*Computer Science*)
 - Scienza dell'informazione
 - Informazione + automatica
 - Scienza e tecnica dell'elaborazione dei dati e, genericamente, del trattamento automatico dell'informazione (Zingarelli)
 - Scienza del trattamento razionale, specialmente per mezzo di macchine automatiche, dell'informazione, considerata come supporto alla conoscenza umana e alla comunicazione (Academie Francaise).
- **Elaboratore elettronico (o “computer” o “calcolatore”):**

è uno **strumento** per la rappresentazione, la memorizzazione e l'elaborazione delle informazioni.



COS'È L'INFORMATICA?

- **INFORMATICA** = Scienza della rappresentazione e dell'elaborazione dell'informazione
 - L'informatica studia le caratteristiche dell'informazione e i modi di usarla, immagazzinarla, trasportarla e manipolarla in modo automatico.
- L'informatica ha due anime:
 - **tecnologica**: i calcolatori elettronici e i sistemi che li utilizzano;
 - **metodologica**: i metodi per la soluzione di problemi e la gestione delle informazioni.



HARDWARE E SOFTWARE

- La prima decomposizione di un calcolatore è relativa alle seguenti macro-componenti
 - **Hardware**

la struttura fisica del calcolatore, costituita da componenti elettronici ed elettromeccanici
 - **Software**

l'insieme dei programmi che consentono all'hardware di svolgere dei compiti utili



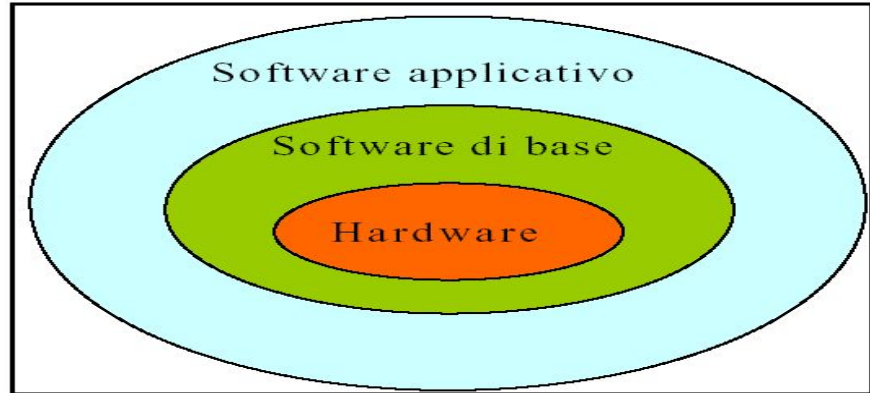
HARDWARE

- ◆ Elaborazione
- ◆ Memorizzazione
- ◆ Trasferimento
- ◆ Controllo

SOFTWARE

Programmi che vengono eseguiti dal sistema:

- **Software di Base** (tra cui il sistema operativo)
- **Software applicativo**





SOFTWARE E MACCHINE VIRTUALI

- L'hardware è l'unica **macchina reale**, mentre i vari strati software corrispondono a macchine virtuali
 - Le operazioni (chiamate **istruzioni**) che l'hardware sa eseguire direttamente costituiscono il **linguaggio macchina** del calcolatore
 - Le istruzioni del linguaggio macchina sono molto semplici, ma il calcolatore può eseguirle in modo molto efficiente
- Il software ha lo scopo di mostrare ai suoi utenti il calcolatore come una **macchina virtuale** (non esistente fisicamente), più semplice da usare rispetto all'hardware sottostante



SOFTWARE E MACCHINE VIRTUALI

- **Macchine virtuali**
 - Semplificano la comunicazione fra uomo e hardware
 - Le diverse macchine e i relativi insiemi di operazioni sono via via più astratti: più vicini alla logica dell'utente e più lontani dalla logica del calcolatore come dispositivo elettronico
 - Alla fine, comunque, l'unico responsabile dell'esecuzione del software è l'hardware disponibile
- Il **software di base** ha lo scopo di mostrare all'utente il calcolatore come una macchina virtuale (più semplice da gestire e programmare rispetto all'hardware utilizzato)
- Il **software applicativo** mostra all'utente il calcolatore come una macchina virtuale utilizzabile per la risoluzione di problemi

RAPPRESENTAZIONE DELLE INFORMAZIONI



RAPPRESENTAZIONE DELLE INFORMAZIONI

- Si utilizzano standard internazionali per risolvere problemi di compatibilità tra calcolatori di tipo e marca diversi
- Vedremo brevemente:
 - Codifica di numeri
 - Operazioni tra numeri attraverso circuiti logici
 - Codifica di caratteri
 - Codifica di dati multimediali



CODIFICA DEI NUMERI

- Gli elaboratori utilizzano la codifica binaria (cioè con 0 e 1) dell'informazione
- Perché solo due simboli?
 - differenti tensioni elettrici, polarità magnetiche, ...
 - Riduce errori (ad es. causati da rumore nei segnali)
- Unità elementare di informazione: bit
interpretato come 0 o come 1
- Unità derivata: byte = 8 bit



CODIFICA DEI NUMERI

- Esempio: 13 può essere espresso in funzione delle potenze di 2 come:

$$13 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$$

- Cioè può essere rappresentato dalla sequenza di bit

1 1 0 1



CODIFICA DEI NUMERI

- Quindi
 - Numero = sequenza di bit (codifica in base 2)
 - Con K bit si rappresenta al massimo $2^K - 1$
- Esempi:
 - 2 = sequenza 1 0
 - 3 = sequenza 1 1
 - 4 = sequenza 1 0 0
 -



OPERAZIONI TRA NUMERI

Ad esempio: somma

$$5 = 1 \times 4 + 0 \times 2 + 1 \times 1$$

$$2 = 0 \times 4 + 1 \times 2 + 0 \times 1$$

$$7 = 1 \times 4 + 1 \times 2 + 1 \times 1$$

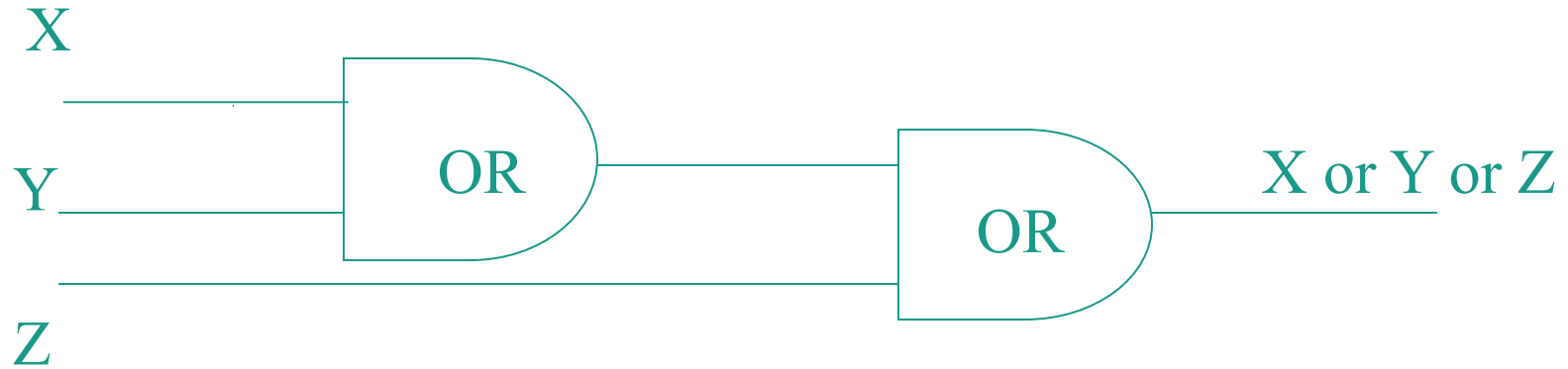
Esprimibile come operazione tra bit



OPERAZIONI LOGICHE

- Operazioni logiche: AND, OR, NOT, ...
 - 0 = falso, 1=vero
 - $X \text{ AND } Y = 1$ sse $X=1$ e $Y=1$
 - $X \text{ OR } Y = 1$ sse $X=1$ oppure $Y=1$
 - $\text{NOT } X = 1$ sse $X = 0$
- Circuiti logici: composizione degli operatori che trasformano ingressi (input) in uscite (output)
- Cioè un circuito definisce una funzione
 - f: input \Rightarrow output
- Input, output = sequenze di bit

CIRCUITI LOGICI





RIASSUMENDO...

- Numeri = codificati come sequenze di bit
- Operazioni = operazioni bit a bit
- Funzioni complesse = circuiti
- Hardware = implementazione fisica dei circuiti logici
- Hardware = fornisce operazioni primitive che vengono utilizzate per definire applicazioni attraverso programmi



CODIFICA DI ALTRI TIPI DI NUMERI

- Reali: si utilizzano codifiche quali
 - Floating point
 - Fixed point
- Interi: si utilizzano codifiche quali
 - Bit di segno
 - Complemento a 1 e a 2



CODIFICA DI CARATTERI

- Codifica **ASCII**: associando un simbolo dell'alfabeto ad ogni numero possiamo codificare tutte le lettere
 - a-z A-Z 0-9usando 7 bit (cioè in un byte)!!
- Esempio: 00000101 rappresenta la lettera 'c'



CODIFICA DI DATI MULTIMEDIALI

- Lettere e numeri non costituiscono le uniche informazioni utilizzate dagli elaboratori ma si diffondono sempre di più applicazioni che usano ed elaborano anche altri tipi di informazione:
 - diagrammi
 - immagini
 - suoni
- Spesso in questi casi si parla di applicazioni di tipo multimediale

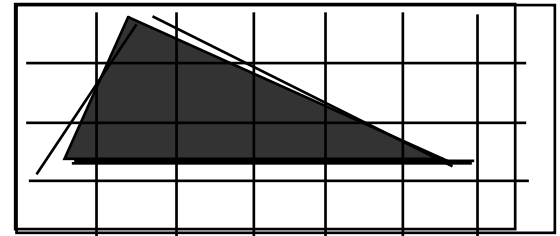
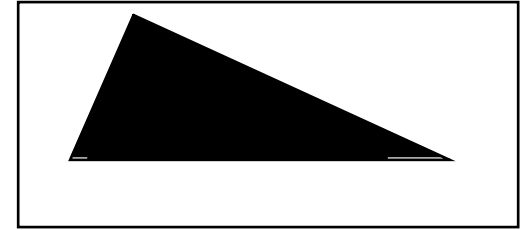


CODIFICA DI IMMAGINI

- Esistono numerose tecniche per la memorizzazione digitale e l'elaborazione di un'immagine
- Immagini = sequenze di bit!
- L'immagine viene digitalizzata cioè rappresentata con sequenze di pixel
- Ogni pixel ha associato un numero che descrive un particolare colore (o tonalità di grigio)
- Inoltre si mantengono la dimensione, la risoluzione (numero di punti per pollice), e il numero di colori utilizzati

CODIFICA DI IMMAGINI

- Consideriamo un'immagine in bianco e nero, senza ombreggiature o livelli di chiaroscuro
- Suddividiamo l'immagine mediante una griglia formata da righe orizzontali e verticali a distanza costante





CODIFICA DI IMMAGINI

- Ogni quadratino derivante da tale suddivisione prende il nome di **pixel** (picture element) e può essere codificato in binario secondo la seguente convenzione:
 - Il simbolo “0” viene utilizzato per la codifica di un pixel corrispondente ad un quadratino bianco (in cui il bianco è predominante)
 - Il simbolo “1” viene utilizzato per la codifica di un pixel corrispondente ad un quadratino nero (in cui il nero è predominante)



CODIFICA DI IMMAGINI

Poiché una sequenza di bit è lineare, si deve definire una convenzione per ordinare la griglia dei pixel in una sequenza

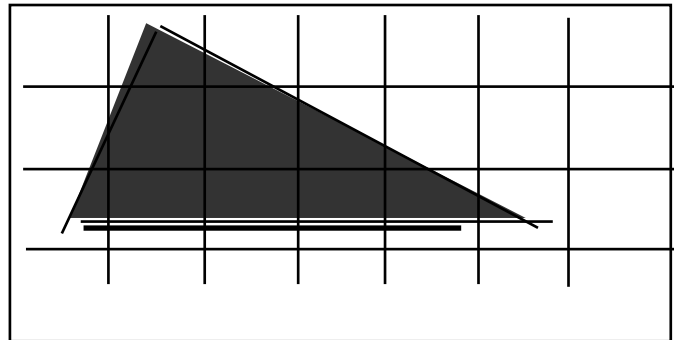
Hp: assumiamo che i pixel siano ordinati dal basso verso l'alto e da sinistra verso destra

0	1	0	0	0	0	0
0 ²²	1 ²³	1 ²⁴	0 ²⁵	0 ²⁶	0 ²⁷	0 ²⁸
0 ¹⁵	1 ¹⁶	1 ¹⁷	1 ¹⁸	1 ¹⁹	0 ²⁰	0 ²¹
0 ⁸	0 ⁹	0 ¹⁰	0 ¹¹	0 ¹²	0 ¹³	0 ¹⁴
1	2	3	4	5	6	7

CODIFICA DI IMMAGINI

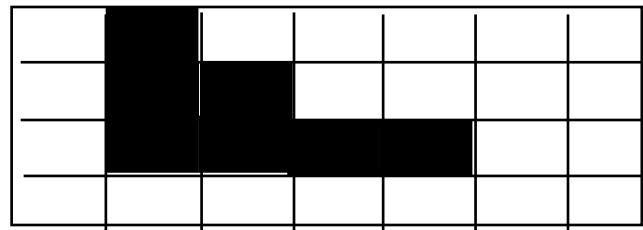
La rappresentazione della figura sarà data dalla stringa binaria

0000000 0111100 0110000 0100000



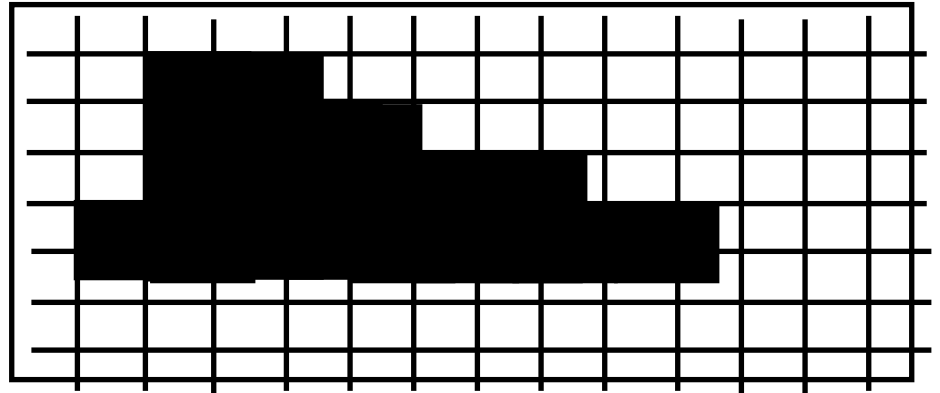
CODIFICA DI IMMAGINI

- Non sempre il contorno della figura coincide con le linee della griglia: nella codifica si ottiene un'approssimazione della figura originaria
- Se riconvertiamo la stringa **0000000011110001100000100000** in immagine otteniamo



CODIFICA DI IMMAGINI

- La rappresentazione sarà più fedele all'aumentare del numero di pixel, ossia al diminuire delle dimensioni dei quadratini della griglia in cui è suddivisa l'immagine





CODIFICA DI IMMAGINI: TONI DI GRIGIO

- Le immagini in bianco e nero hanno delle sfumature (diversi **livelli di intensità di grigio**), ma assegnando un bit ad ogni pixel è possibile codificare solo immagini senza livelli di chiaroscuro
- Per codificare le immagini con diversi livelli di grigio si usa la stessa tecnica: per ogni pixel si stabilisce il livello medio di grigio cui viene assegnata convenzionalmente una rappresentazione binaria
- Per memorizzare un pixel non è più sufficiente un solo bit.
 - ad esempio, con 4 bit possiamo rappresentare $2^4=16$ livelli di grigio, mentre con 8 bit ne possiamo distinguere $2^8=256$, ecc.



CODIFICA DI IMMAGINI A COLORI

- Analogamente possiamo codificare le immagini a colori. In questo caso si tratta di individuare un certo numero di sfumature di colore differenti e di codificare ogni sfumatura mediante un'opportuna sequenza di bit
- La rappresentazione di un'immagine mediante la codifica dei pixel, viene chiamata **codifica bitmap**



CODIFICA DI IMMAGINI A COLORI

- Il numero di byte richiesti dipende dalla risoluzione e dal numero di colori che ogni pixel può assumere
 - i monitor utilizzano risoluzioni di 640×480 , 1024×768 , oppure 1280×1024 ed un numero di colori per pixel che va da 256 fino a 16 milioni
- Per distinguere **256** colori sono necessari 8 bit per la codifica di ciascun pixel:
 - la codifica di un'immagine formata da 640×480 pixel richiederà 2457600 bit (307200 byte)
- Esistono delle tecniche di **compressione** delle informazione che consentono di ridurre drasticamente lo spazio occupato dalle immagini



CODIFICA DI FILMATI

- Immagini in movimento: memorizzazione mediante sequenze di fotogrammi (sono necessarie delle tecniche per ottimizzare tale memorizzazione)
- Sono sequenze di immagini compresse: (ad esempio si possono registrare solo le variazioni tra un fotogramma e l'altro)
- Esistono vari formati (compresi i suoni):
 - *mpeg* (il piu' usato)
 - *avi* (microsoft)
 - *quicktime* (apple)
 - *mov*
- E' possibile ritoccare i singoli fotogrammi



CODIFICA DI SUONI

- L'onda sonora viene misurata (campionata) ad intervalli regolari
- Minore e l'intervallo di campionamento e maggiore e la qualità del suono
- CD musicali: 44000 campionamenti al secondo, 16 bit per campione.
- Alcuni formati:
 - .mov, .wav, .mpeg, .avi,
 - formato **midi** usato per l'elaborazione della musica al PC

IL CALCOLATORE



HARDWARE

- ◆ Elaborazione
- ◆ Memorizzazione
- ◆ Trasferimento
- ◆ Controllo

ELABORAZIONE

- ◆ Le istruzioni del **linguaggio macchina** corrispondono ad operazioni elementari di elaborazione
 - operazioni aritmetiche
 - operazioni relazionali (confronto tra dati)
 - operazioni logiche
 - ...
- ◆ Un calcolatore sa svolgere poche tipologie di operazioni elementari ma in modo molto efficiente
 - un calcolatore può eseguire decine o centinaia di milioni di istruzioni del linguaggio macchina al secondo
- ◆ L'elaborazione è svolta dall'**unità aritmetico-logica**, che è un componente dell'unità centrale di elaborazione

MEMORIZZAZIONE

- ◆ La memoria centrale contiene
 - **dati**, che rappresentano informazioni di interesse
 - **programmi**, per l'elaborazione dei dati
- ◆ Organizzazione
 - è organizzata in celle
 - a ciascun byte è associato un **indirizzo**, che lo identifica
 - una **word** è un gruppo di byte (capacità del bus dati o dimensione di un registro della CPU)
- ◆ Operazioni
 - **scrittura**, memorizzazione di un valore in un byte/word
 - **lettura**, accesso al valore memorizzato in un byte/word

TRASFERIMENTO

- ◆ Obiettivo: permettere lo scambio di informazioni tra le varie componenti funzionali del calcolatore
 - trasferimento dei dati e delle informazioni di controllo
- ◆ Due possibili soluzioni
 - collegare ciascun componente con ogni altro componente
 - collegare tutti i componenti ad un unico canale (**bus**)
- ◆ L'utilizzo di un bus favorisce la modularità e l'espandibilità del calcolatore



CONTROLLO

- ◆ Il coordinamento tra le varie parti del calcolatore è svolto dall'**unità di controllo**
 - è un componente dell'unità centrale di elaborazione
 - ogni componente del calcolatore esegue solo le azioni che gli vengono richieste dall'unità di controllo
- ◆ il controllo consiste nel coordinamento dell'esecuzione temporale delle operazioni
 - sia internamente all'unità di elaborazione sia negli altri elementi funzionali
 - il controllo avviene in modo sincrono rispetto alla scansione temporale imposta dall'orologio di sistema (**clock**)



ARCHITETTURA

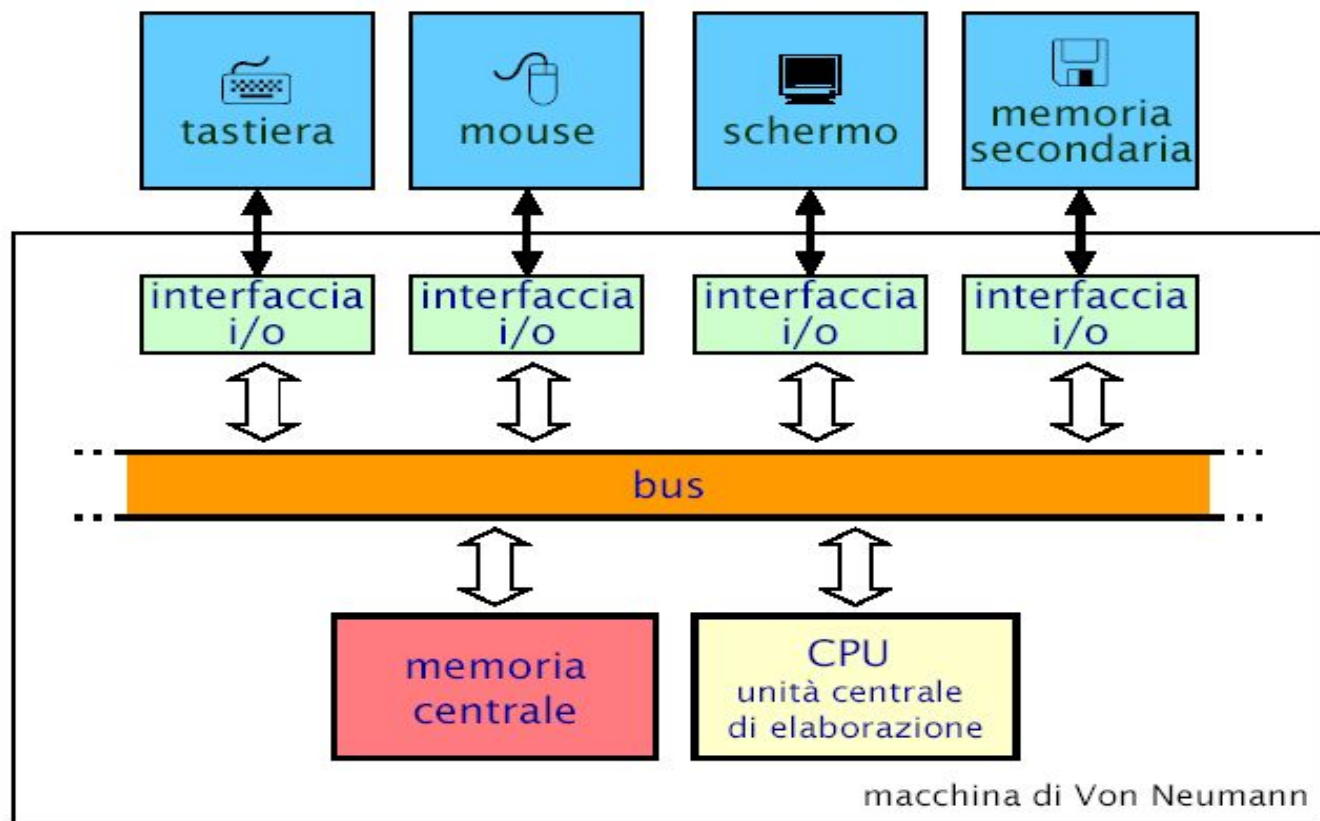
- ◆ L'architettura dell'**hardware** di un calcolatore reale è molto complessa
- ◆ La **macchina di Von Neumann** è un modello semplificato dei calcolatori moderni
 - **Von Neumann** progettò, verso il 1945, il primo calcolatore con programmi memorizzabili anziché codificati mediante cavi e interruttori

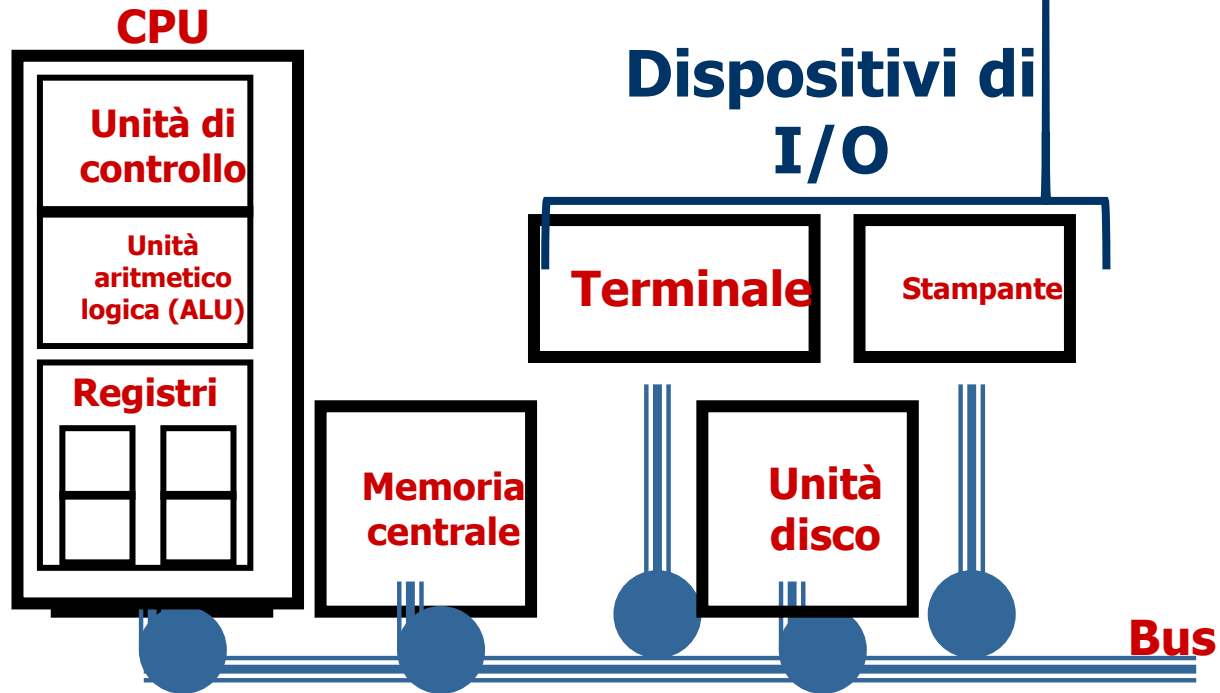


MACCHINA DI VON NEUMANN

E' composta da 4 tipologie di componenti funzionali:

- ◆ unità centrale di elaborazione (CPU)
 - esegue istruzioni per l'elaborazione dei dati
 - svolge anche funzioni di controllo
- ◆ memoria centrale
 - memorizza e fornisce l'accesso a dati e programmi
- ◆ interfacce di ingresso e uscita
 - componenti di collegamento con le periferiche del calcolatore
- ◆ bus
 - svolge la funzionalità di trasferimento di dati e di informazioni di controllo tra le varie componenti funzionali







BUS di sistema

- ◆ Il bus trasporta dati, indirizzi e comandi
- ◆ Componenti del bus (sottogruppi di linee):
 - **Bus dati (data bus)**
 - **Bus indirizzi (address bus)**
 - **Bus comandi (command bus)**
- ◆ **Bus dati (data bus)**
 - Serve per trasferire dati
 - tra la memoria centrale ed il registro dati (MDR) della CPU
 - tra periferiche e CPU (o memoria centrale)
 - Bidirezionale



BUS di sistema

◆ **Bus indirizzi (address bus)**

- Serve per trasmettere il contenuto del registro indirizzi (MAR) alla memoria (o ad una periferica)
 - si seleziona una cella per successive operazioni di lettura o scrittura
- Unidirezionale

◆ **Bus comandi (command bus)**

- Serve per inviare comandi
 - verso la memoria (es: lettura o scrittura)
 - o verso una periferica (es. stampa verso la stampante / interfaccia)
- Unidirezionale

LA PROGRAMMAZIONE

LA PROGRAMMAZIONE

È l'attività con cui si predispone l'elaboratore ad eseguire un *particolare insieme di azioni* su una *particolare tipologia di dati*, allo scopo di *risolvere un problema*.





LA PROGRAMMAZIONE

- Alcune domande fondamentali:
 - *Quali istruzioni esegue (cosa può fare) un elaboratore?*
 - *Quali problemi può risolvere un elaboratore?*
 - *Esistono problemi che un elaboratore non può risolvere?*
 - *Che ruolo ha il linguaggio di programmazione?*
- Il problema di fondo
 - *Come si costruisce la soluzione a un problema?*
 - *Qual è il giusto “punto di partenza” per pensare la soluzione a un problema?*
 - *Quali metodologie e tecniche usare?*




I PROBLEMI

I problemi affrontati dalle applicazioni informatiche sono di natura *molto varia*:

- Trovare il maggiore fra due numeri
- Dato un elenco di nomi e numeri di telefono, trovare il numero di una data persona
- Dati a e b , risolvere l'equazione $ax+b=0$
- Stabilire se una parola precede alfabeticamente un'altra
- Ordinare un elenco di nomi
- Creare, modificare e alterare suoni
- Analizzare, riconoscere e modificare immagini
- Gestione delle aziende (private e pubbliche)
- Supportare operazioni di commercio elettronico

I PROBLEMI



La descrizione del problema non indica direttamente (in genere) un modo per ottenere il risultato voluto


Differenza tra *specificazione di un problema* e *specificazione del processo di risoluzione*

Risoluzione di un problema

processo che:

- dato un *problema*
 - individuato un opportuno *metodo risolutivo (algoritmo)*
- trasforma i dati iniziali nei corrispondenti risultati finali.

ALGORITMI



***Algoritmo:** Sequenza finita di passi che risolve in un tempo finito un problema.*

Esempi di algoritmi:

- Istruzioni di montaggio
- Preparazione del caffè
- Prelievo bancomat
- Preparazione di un ricetta
- Calcolo del massimo comune divisore tra due interi



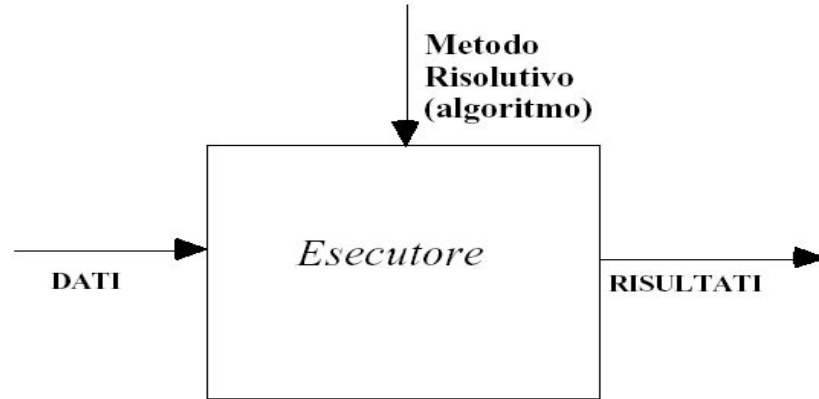
La parola **algoritmo** deriva dal nome di un autore scientifico persiano del IX secolo

- **Abu Ja'far Mohammed ibn Musa al-Khowarizmi** scrisse, circa nell'825, il trattato “*Kitab al jabr w'al-muqabala*” (forse *regole di trasposto e semplificazione*) dove descrisse delle regole per la semplificazione delle equazioni.
- **algebra** deriva da **al jabr** (parte del titolo del trattato)
- **algoritmo** deriva da **Khowarizmi** (ultima parte del nome dell'autore, indicante la città di nascita)

Il termine originario era *agorismo*, trasformato in *algoritmo* per analogia con *aritmetica*.

- **Esecuzione:** L'esecuzione delle azioni *nell'ordine specificato dall'algoritmo* consente di ottenere, a partire dai dati di ingresso, i risultati che risolvono la particolare **istanza** il problema.

Esecutore: una *macchina astratta* capace di eseguire le azioni specificate dall'algoritmo.



ALGORITMI: PROPRIETÀ FONDAMENTALI



- **Eseguibilità:** ogni azione deve essere *eseguibile* da parte dell'esecutore dell'algoritmo in un tempo finito
- **Non-ambiguità:** ogni azione deve essere *univocamente interpretabile* dall'esecutore
- **Finitezza:** il numero totale di azioni da eseguire, per ogni insieme di dati di ingresso, deve essere finito.

Non si può risolvere un problema senza prima fissare un insieme di “azioni”, di “mosse elementari” possibili per l'esecutore.

Bisogna conoscerne le caratteristiche, le mosse che sa eseguire ed il linguaggio che sa capire

ALGORITMO: PROPRIETÀ FONDAMENTALI

Dunque, un algoritmo deve essere:

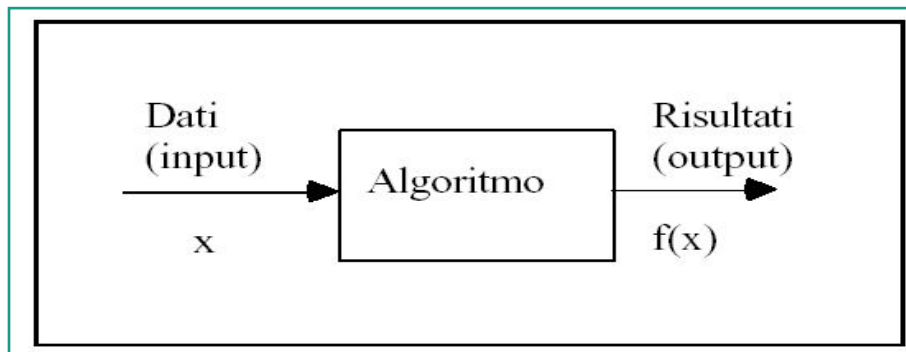
- *applicabile a qualsiasi insieme di dati di ingresso* appartenenti al **dominio di definizione dell'algoritmo**;
- costituito da operazioni appartenenti ad un determinato **insieme di operazioni fondamentali**
- costituito da **regole non ambigue**, cioè interpretabili in modo **univoco** qualunque sia l'esecutore (persona o “macchina”) che le legge

Altre proprietà desiderabili

- generalità
- determinismo
- **efficienza**

ALGORITMI EQUIVALENTI

- In generale un algoritmo può essere visto come una **funzione** da un dominio di ingresso (*input*) ad dominio di uscita (*output*)





ALGORITMI EQUIVALENTI

- Due algoritmi si dicono **equivalenti** quando:
 - hanno stesso dominio di ingresso e stesso dominio di uscita;
 - in corrispondenza degli stessi valori nel dominio di ingresso producono gli stessi valori nel dominio di uscita.



PROBLEMI NON RISOLVIBILI

Ci sono problemi non risolvibili da nessun modello di calcolo reale o astratto

- Esempio 1: predire il valore delle azioni
- Esempio 2: predire se l'Inter (o la Juventus) vincerà il campionato.

RISOLUZIONE DI PROBLEMI CON L'ELABORATORE ELETTRONICO



- Ogni elaboratore è una macchina in grado di eseguire *azioni* elementari su *dati*
- L'esecuzione delle azioni elementari è richiesta all'elaboratore tramite comandi chiamati *istruzioni*
- Le istruzioni sono espresse attraverso *frasi* di un opportuno *linguaggio di programmazione*
- Un *programma* non è altro che la formulazione testuale di un algoritmo in un linguaggio di programmazione

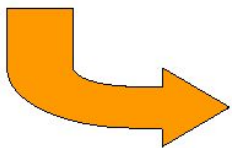
ALGORITMI E PROGRAMMI

- **Algoritmo**

sequenza finita di passi che risolve in un tempo finito un problema.

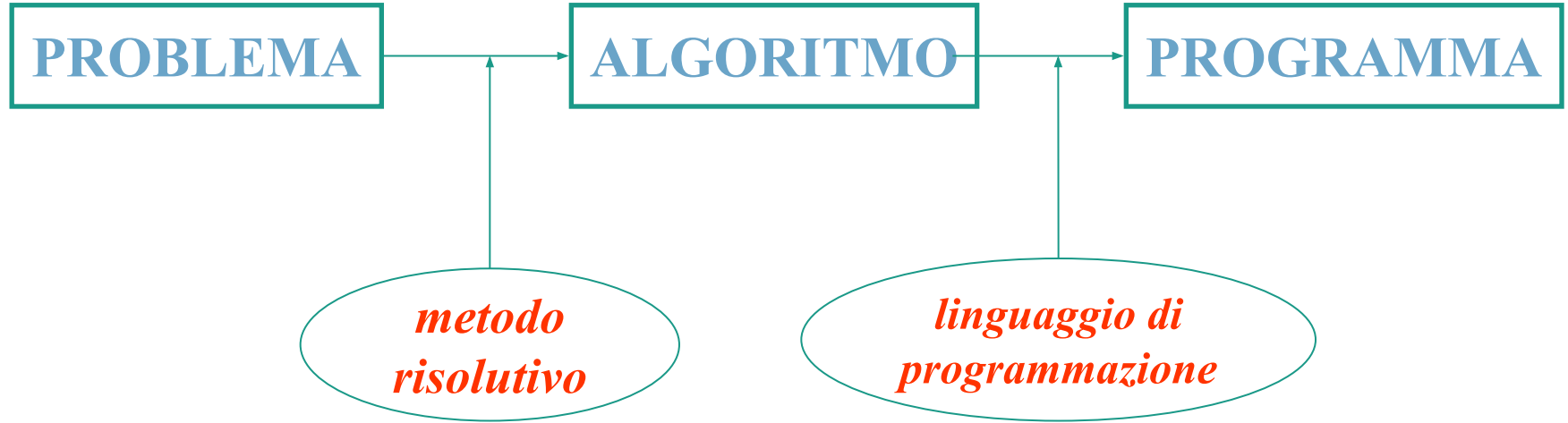
- **Codifica**

fase di scrittura di un algoritmo attraverso un insieme ordinato di frasi (“istruzioni”), scritte in un qualche **linguaggio di programmazione**, che specificano le **azioni** da compiere.



Programma: Testo scritto in accordo alla sintassi e alla semantica di un linguaggio di programmazione

ALGORITMI E PROGRAMMI



LINGUAGGI: SINTASSI E SEMANTICA



- **Sintassi:** l'insieme delle regole che consentono di scrivere parole e frasi riconoscibili come appartenenti ad un determinato linguaggio.
 - collegamento ordinato delle parole nel discorso
- **Semantica :** la disciplina che studia il significato delle parole e delle frasi.



LINGUAGGI AD ALTO LIVELLO

È **opportuno** impostare la soluzione di un problema a partire dalle “mosse elementari” del **linguaggio macchina**?

- SI, per risolvere il problema *con efficienza*
 - NO, se la macchina di partenza ha mosse di livello *troppo basso* (difficile progettare un algoritmo)
- ↓

Linguaggi di Programmazione ad Alto Livello (di astrazione)

- le istruzioni corrispondono ad operazioni più complesse
- esempi: *Pascal, Basic, C, C++, Java, Python*



ESEMPIO: calcolo del MCD

Problema: calcolo del Massimo Comun Divisore (MCD) fra due interi M ed N

Algoritmo n° 1

1. Calcola l'insieme A dei divisori di M
2. Calcola l'insieme B dei divisori di N
3. Calcola l'insieme C dei divisori comuni $= A \cap B$
4. Il risultato è il massimo dell'insieme C

ESEMPIO: calcolo del MCD



Algoritmo di Euclide

Dati due interi M e N

1. Dividi M per N , e sia R il resto della divisione;
2. Se $R=0$ allora termina e il Massimo Comune Divisore è N ;
3. Assegna a M il valore di N ed a N il valore del resto e torna al punto 1.



Domande?

masciadri.andrea@gmail.com

Ora provate voi..



References

- Gianluigi Folino, CORSO DI INTRODUZIONE ALL' INFORMATICA A.A 2004/2005
- Carmela Comito, FONDAMENTI DI INFORMATICA