

FONDAMENTI DI INFORMATICA

INTRODUZIONE AL CORSO E PRIME NOZIONI

Prof. Alfredo Accattatis

(accattatis@ing.uniroma2.it)

Tutor:

Prof. Vittorio Emanuele Calafiore

(vcalafio@gmail.com)

Informazioni

- **INGEGNERIA ENERGETICA/MECCANICA**
- Aula “fisica” + bonus registrazione lezione (se funziona)
- Ingegnere Informatico/Elettronico + Ph.D.
- Esperienza : accademica, ditte private e pubbliche, professione (es. perizie, impianti fotovoltaici, software...)
- ...Siamo (saremo) colleghi!
- Il corso in aula (fisica, virtuale o registrata) aggiunge, rispetto a libri e slide, elementi presi da esperienza “sul campo” (mondo “reale” degli Ingegneri)
- Possibilità di lezione extra su argomenti correlati al lavoro

Modalità svolgimento lezioni

- Possibilità di fare ricevimento tramite mail, messaggio e se necessario si fissa un appuntamento in CALL (TEAMS)
- Il corso prevede il rilascio del materiale su **didatticaweb** (<https://didattica.uniroma2.it/home/accedi>) dove troverete le slide ed altro materiale *DI VOLTA IN VOLTA* e in simultanea su canale TEAMS, se abilitato:
- ACCATTATIS-8037706-
FONDAMENTI_DI_INFORMATICA_5

Ricapitolando: Coordinate (Didatticaweb e Teams)

- Classe virtuale: ACCATTATIS-8037706-FONDAMENTI_DI_INFORMATICA_5
- <https://didattica.uniroma2.it/home/accedi>

Iscrizione al corso

❑ E' stata abilitata la sezione iscrizione al corso sulla piattaforma DELPHI (delphi.uniroma2.it),

❑ Grazie a chi si è iscritto!

❑ La richiedo (anche) per fini statistici, ed in caso di lezioni online o ricevimento online diventa importante - e imprescindibile per iscriversi agli esami.

Dove trovare il materiale

- Sito ufficiale (Didatticaweb, Teams)
 - Informazioni, slide
 - Libri di testo (li vedremo a breve)
 - Gradito feedback
 - <http://didatticaweb.uniroma2.it>

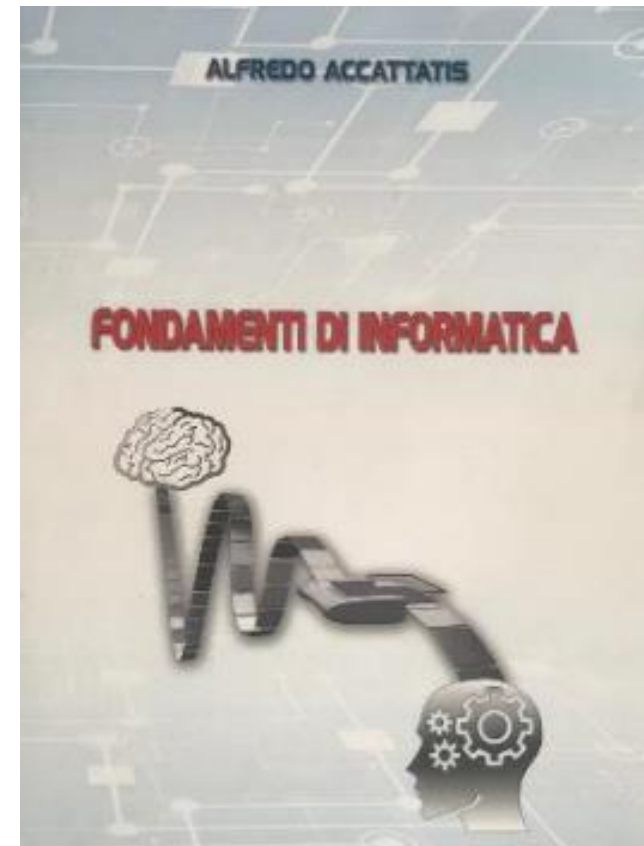


- Amici
- Forum
- Varie...



Materiale per studiare

- Slide e Libro di testo
 - **Slide** rilasciate dopo ogni lezione
 - Testo: Basi di Informatica(Accattatis)
 - Testo: Comprende Esercizi (Accattatis)
 - Libreria TEXMAT, Via di Tor Vergata 93/95
 - <https://www.texmat.it/>
 - **Rinuncia** ai diritti di autore
- Tool di programmazione:
 - Matlab - <http://www.mathworks.com>
 - Octave - <http://www.gnu.org/software/octave/>



Nota: le slide non sostituiscono il libro di testo!

Alfredo Accattatis è laureato in Ingegneria Elettronica, Ingegneria Informatica e Dottorato in Microelettronica e Telecomunicazioni presso l'Università di Roma "Tor Vergata"; è abilitato all'esercizio della professione presso l'Università di Roma "La Sapienza" ed è iscritto all'albo degli Ingegneri di Roma. È laureato in Psicologia presso l'Università degli studi "Niccolò Cusano". Ha lavorato presso Eurolab s.r.l. con il professor Armando Bellini (progettazione hardware e software di azionamenti industriali a orientamento di campo, linguaggi usati Pascal, C++, Assembly microcontrollori e DSP); in Gesinf s.r.l. (progettazione Sistemi Informativi in Delphi, C e Pascal); in Alenia Difesa S.p.A. (reparto Sistemi Avionici ed equipaggiamenti, progettazione software avionico per aerei da caccia in ADA, C++); Eciesson S.p.A. (ricerca e sviluppo in ambito DSP, linguaggi usati Assembly e C++). Ha collaborato con le riviste Nuova Elettronica, Fare Elettronica, Elettronica In (produzione di software, hardware e articoli). Attualmente è Funzionario Informatico e Capo Progetto presso Inps D.C.T.I.I. (Direzione Centrale Tecnologie Informatiche ed Innovazione), inserito nell'area Sicurezza Informatica e successivamente nell'area Esercizio Applicazioni (gestione CED con turni H24); linguaggi usati C++, REXX e sistemi operativi Z/OS, Windows, Linux. Nel 2012 è stato docente di Metodologie di Misure Interne a Sistemi Virtuali presso l'Università di Roma "Tor Vergata" e del corso integrativo di Strumentazione Virtuali per le Verifiche di Sicurezza Elettrica nel corso di Sicurezza Elettrica del prof. Roberto Lojano; dall'anno accademico 2015/16 ad oggi è docente a contratto del corso Fondamenti di Informatica presso l'Università di Roma "Tor Vergata". È consulente presso tribunali civili e penali con all'attivo numerose perizie come CTU e CTP. È esperto di Elaborazione Numerica dei Segnali (DSP) e autore del software di Misure Virtuali "Visual Analyser" (www.sillammsoft.org). Dal 2002 è Progettista di Impianti Fotovoltaici a Isola e in Rete e autore del software "Solar Patrol" per la gestione di impianti fotovoltaici ad isola; è autore del libro divulgativo "Cambiamento climatico (e possibili soluzioni: auto elettriche, fotovoltaico, nucleare)".

€ ??,??

ALFREDO
ACCATTATISBASIDIINFORMATICA
CON MATLAB ED ESERCIZI

TEXMAT

ALFREDO ACCATTATIS



TEXMAT

Novità:

- Da Complementi (rivisti)
 - Programmazione Object Oriented
 - Ricorsione (cenni veloci)
 - Esercizi reali d'esame commentati
- Nuovi capitoli
 - Sistemi operativi
 - Complessità
 - Importanti modifiche nei capitoli importati: dimensione e risoluzione figure, refusi, interi paragrafi resi più chiari, miglioramento delle note

Come studiare....

- Il testo è sovrabbondante...
- Il corso è da 6 (sei) crediti e comprende MOLTI argomenti
- Soluzione:

Usate le slide come linea guida ed il testo per «comprendere» le slide...e quando richiesto fate gli esercizi!!

All'esame verrà chiesto SOLO quanto fatto a lezione (slide)

Premessa:

Il corso di fondamenti di Informatica è per voi *strumentale*, ossia l'informatica è solo un **mezzo** e non un fine.

Lo scopo del corso è dunque quello di fornire gli strumenti teorici di base e principalmente la capacità di scrivere dei programmi con un linguaggio di programmazione

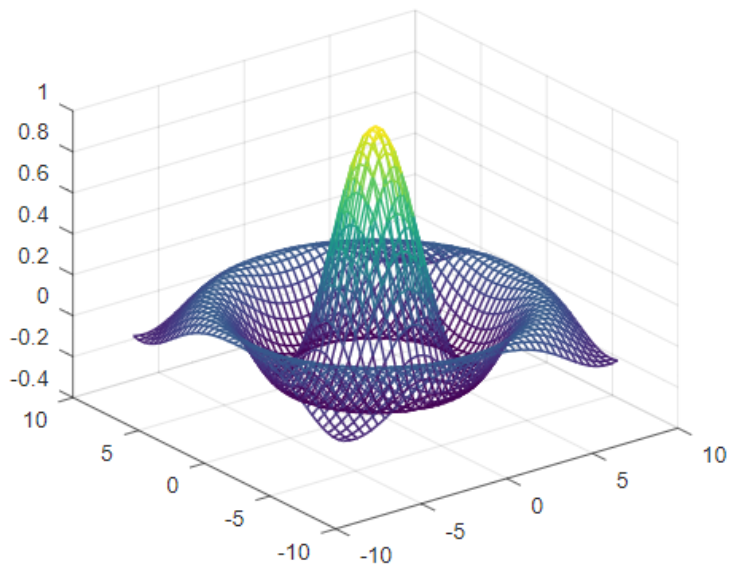
Perché Matlab/Octave

- Matlab è più di un semplice linguaggio. Esso è un “pacchetto”, ossia un insieme di programmi associati ad un linguaggio di programmazione che permette di *automatizzare gli algoritmi* (e questo lo fa ogni linguaggio di programmazione) ma consente di andare molto oltre (esempio: simulazione di sistemi, compilazione, grafica, programmazione schede dedicate etc..)
- Matlab è un linguaggio ad “*alto livello*”
- Permette di tradurre in pratica i concetti teorici appresi durante le lezioni di Fondamenti di Informatica.
- Questo NON è un corso di “solo” MATLAB
- Insegna i principi di base di TUTTI i linguaggi

OCTAVE

- Octave con GUI (attuale 9.4.0):

1. Per windows/Linux si puo' scaricare da <https://octave.org/download>



Scientific Programming Language

- Powerful mathematics-oriented syntax with built-in 2D/3D plotting and visualization tools
- Free software, runs on GNU/Linux, macOS, BSD, and Microsoft Windows
- Drop-in compatible with many Matlab scripts

Download

Documentation

MATLAB: come ottenerlo

- <https://it.mathworks.com/academia/tah-portal/universita-degli-studi-di-roma-tor-vergata-31106316.html>
- <https://docs.ccd.uniroma2.it/matlab/>
- <https://www.mat.uniroma2.it/Calcolo/Matlab/MatlabInstall.html>

Esami

L'esame sarà composto da:

- Esame scritto (DUE risposte aperte e TRE esercizi) su tutti gli argomenti affrontati nel corso, tranne quelli esplicitamente esclusi (verrà detto a lezione e comunicato telematicamente): 2025 forse novità
- Nuovo libro comprende esercizi d'esame
- **Emergenza COVID** : esame a quiz, 31 domande con 4 opzioni una corretta e 32 minuti di tempo (TEAMS)

ChatGPT, Gemini etc...

- Ottimo per scrivere codice
- Ma solo come linea guida, dovete fare esercizi!
- NON usare all'esame

Il vostro background

- Sapete cosa è lo ***Spettro di un segnale***?
- Teorema del campionamento?
- Li farete certamente in altri corsi, ma in questo ne daremo comunque una prima introduzione al fine di comprendere meglio problematiche generali sul software. Il teorema del campionamento ha consentito lo sviluppo dell'informatica come la conosciamo ora.

Perché programmare?

- Perché sviluppare questa capacità ?
 - Apprendere l'arte di risolvere un problema tramite un computer è utile in generale nella vita
 - Alla stessa stregua, ed a maggior ragione, sarà utile nella vostra vita professionale
 - Quali computer abbiamo a disposizione?
 - PC, Tablet, Smartphone....

Il telefonino è un computer. Allora non usatelo solo per usare APP preconfezionate. Scrivetevi il software da soli!

Opinioni autorevoli

- Non è solo la mia opinione ma:
 - Obama (US Gov.) ha detto: **“Don’t Just Play on Your Phone, Program It”**



L'informatica va oltre

- Ha influenzato materie apparentemente assai distanti
- Psicologia
- Comportamentismo vs cognitivismo
- L'approccio cognitivista (**Ulrich Neisser**) e poi HIP (Human Information Processing). Approccio computazionale di Marr (visione)
- Sono basati su una modellizzazione del comportamento umano basato sugli stessi paradigmi dell'Informatica e dei Computer, in pratica il modello di *von Neumann*, che studieremo
- La mente opera attivamente sull'informazione proveniente dagli organi sensoriali, trasformata dal soggetto in base ai suoi scopi, aspettative ed esperienze passate (memorizzate) => Comportamento

Psicologia cognitiva

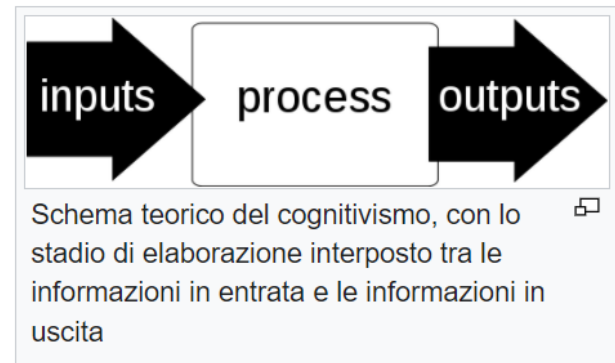
Da Wikipedia, l'enciclopedia libera.

(Reindirizzamento da **Cognitivismo**)

La **psicologia cognitiva**, anche detta **cognitivismo**, è una branca della **psicologia** applicata allo studio dei **processi cognitivi**, teorizzata intorno al 1967 dallo psicologo statunitense **Ulric Neisser**, che ha come obiettivo lo studio dei processi **mentali** mediante i quali le **informazioni** vengono acquisite dal **sistema cognitivo**, elaborate, memorizzate e recuperate.

Indice [nascondi]

- 1 [Descrizione](#)
- 2 [Storia](#)



Descrizione [\[modifica \]](#) [\[modifica wikitesto \]](#)

Essa studia la **mente** come elemento intermedio tra il **comportamento** e l'attività cerebrale prettamente **neurofisiologica**. Il modello di funzionamento è assimilato (**metaforicamente**) a quello di un **software** che elabora informazioni provenienti dall'esterno (*input*), restituendo a sua volta informazioni (*output*) sotto forma di rappresentazione della **conoscenza**, organizzata in reti semantiche e cognitive.

Elaborazione umana dell'informazione

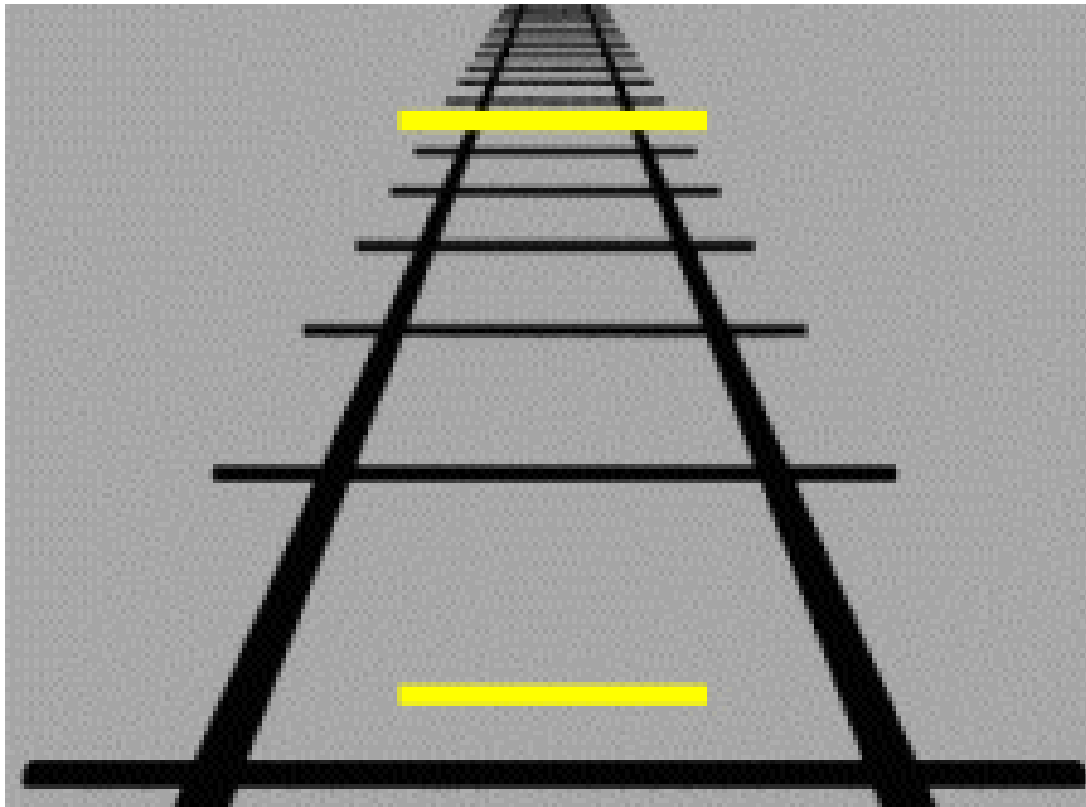
Da Wikipedia, l'enciclopedia libera.

L'elaborazione umana dell'informazione (in inglese *human information processing* o HIP) è un quadro di riferimento teorico che vede la mente umana intervenire sull'informazione proveniente dagli organi sensoriali, trasformandola in base a scopi, aspettative ed esperienze passate del soggetto. Il nome di questo tipo di studio deriva dal titolo del libro scritto nel 1972 da Peter H. Lindsay e [Donald A. Norman](#) *Human information processing: an introduction to psychology*; il campo specialistico su cui si basa è la [psicologia](#).

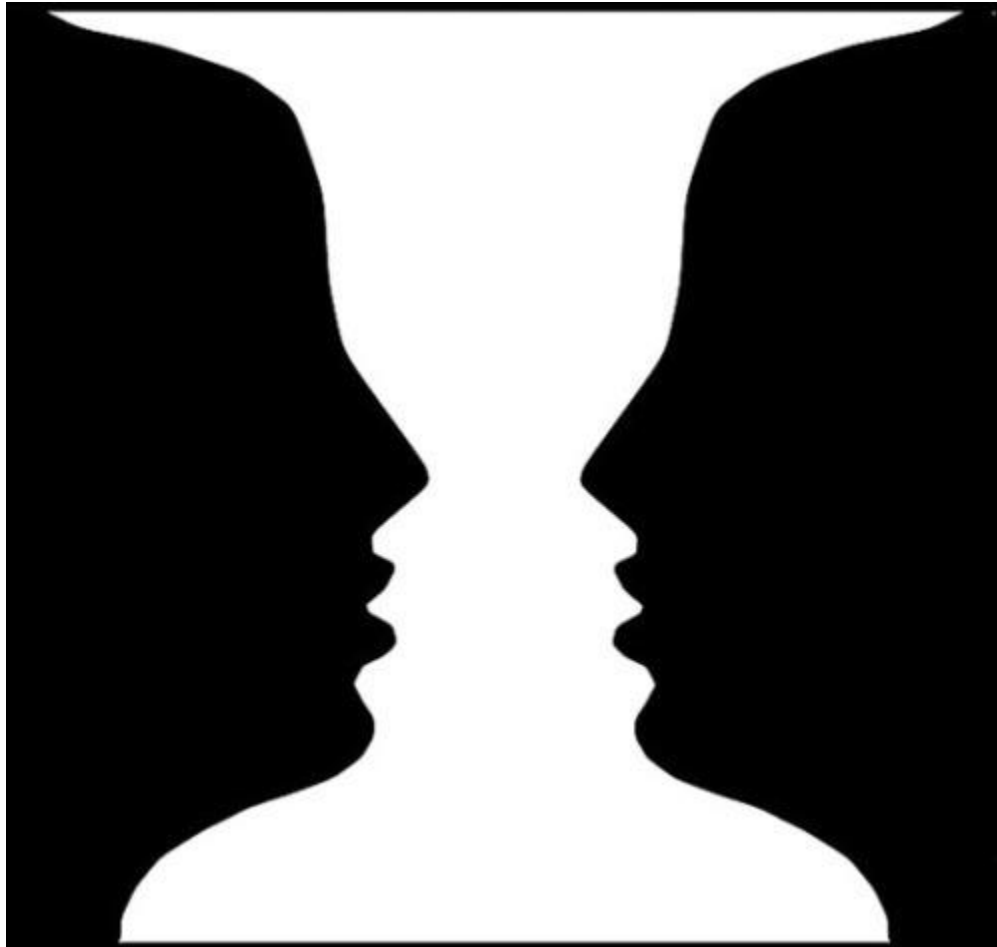
L'informatica va oltre

- Psicologia della percezione
- Esempio: Visione!
- Due grandi scuole di pensiero
 - Percezione diretta (nessuna elaborazione)
 - Percezione indiretta (elaborazione)

Illusione di Ponzo



Viso-vaso di Rubin



Definizione di Informatica (Computer Science)

- Le definizioni sono varie...ma simili....prendiamone una autorevole

Treccani:

- *Scienza che studia l'elaborazione delle informazioni e le sue applicazioni; più precisamente l'Informatica si occupa della **rappresentazione**, **dell'organizzazione** e del trattamento **automatico** della **informazione***

"Computer science is no more about computers than astronomy is about telescopes." (L'informatica non riguarda i computer più di quanto l'astronomia riguardi i telescopi)

E. Dijkstra

Quattro elementi fondamentali

- (1) **Informazione**
 - (2) **Rappresentazione** (dell'informazione)
 - (3) **Organizzazione** (dell'informazione)
 - (4) Trattamento **automatico** (dell'informazione)
 - Algoritmi automatizzati
-
- In questa lezione vedremo principalmente (1) e (2)
 - Concetto di algoritmo, legato a (4)
 - Il (3) lo vedremo meglio più avanti

(1) Informazione

- **Dati** (descrizione elementare e non interpretata della realtà: fatti o eventi rappresentati da simboli)
- **Informazioni** (da un significato ai dati: dato contestualizzato. *Esempio: suono di una sirena*)
 - Scambio di conoscenza tra esseri umani
- **Conoscenza** (uso delle informazioni con l'esperienza, competenza di una specifica materia, in generale «saper fare»)

(2) Rappresentazione dell'Informazione

l'informatica ha a che fare con le “informazioni” ossia quelle entità che ci consentono di effettuare lo “scambio di conoscenza” tra esseri umani. La *rappresentazione* ha a che fare i merito al “**come**” questa informazione viene memorizzata negli elaboratori.

Come fare dunque a rendere comprensibile e memorizzabile al calcolatore il concetto astratto di informazione? La risposta è che dobbiamo *rappresentare* essa per mezzo di simboli *tradotti* in una grandezza fisica disponibile nel mondo (elettronico) del calcolatore. Ossia effettuare quella che viene definita “codifica” dell'informazione

La scelta per i calcolatori è la rappresentazione
tramite numeri in BASE DUE

In particolare per i calcolatori la base scelta è due, dunque i numeri sono rappresentati come sequenze di zeri e uno (BIT). Perché questa scelta? Perché ad esempio non usare la base 10, a noi più familiare?

Per motivi di pratica e tecnologica realizzabilità: distinguere tra due valori di una grandezza fisica (per esempio due livelli di tensione elettrica) è più semplice.

Richiamo: passaggio da base 2 a base 10:

$$d_n 2^n + d_{n-1} 2^{n-1} + \dots + d_1 2^1 + d_0 2^0 = N_{10}$$

$$1001_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 9_{10}$$

Come è possibile far interagire il computer con il "mondo reale" ? Come è possibile far sì che le grandezze del mondo esterno vengano trasferite nel mondo dell'elaboratore?

Provate a pensare:

- Musica
- Immagini
- Filmati
- Voce
- Eccetera (es. temperatura, mappature...)

Sono sempre e solo sequenze di NUMERI
da memorizzare ed elaborare

La risposta è:

Teorema del campionamento!

(Shannon Nyquist)

Lo tratteremo in maggior dettaglio. Tramite esso è possibile tradurre qualsiasi grandezza «del mondo esterno» in un formato comprensibile al mondo degli elaboratori (numeri codificati in binario)

**E' fondamentale ed ha reso
l'Informatica uno strumento universale**

Punto della situazione

- (X) Informazione $\times$
- (X) **Rappresentazione** (dell'informazione)
- (3) **Organizzazione** (dell'informazione)
- (4) Trattamento **automatico** (dell'informazione)
 - Algoritmi automatizzati

(3) Organizzazione

- Lo vedremo meglio quando saremo in una fase più avanzata del corso
- Informazioni e dati possono essere raggruppati in unità complesse che sono considerate atomiche
- Per esempio l'informazione indirizzo:
 - E' composta dal nome di una via, numero e città
 - Ossia è un insieme di numeri, che possiamo organizzare nell'entità atomica «indirizzo»
- Array, Structure, Cell Array, File
- DBMS (archivi di dati)

(4) Trattamento automatico

Algoritmo, soluzione di un problema

Problem solving

L'arte di risolvere i problemi con il
computer

Diamo una definizione generale di algoritmo, rappresentazione
e di esecuzione automatica



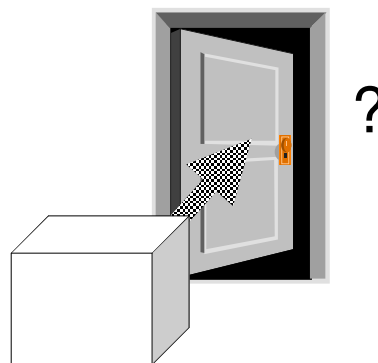
Algoritmo: sequenza di passi elementari
per risolvere un problema

Modalità tipicamente umana per eseguire un «compito»

Esempio tipico: ricetta di cucina!

Esempio. Un problema da risolvere.

- Cosa ci serve per risolvere il problema “astrattamente” ?
 - Una *rappresentazione* delle entità coinvolte (porta, scatola)
 - Una procedura da seguire (basata sulla rappresentazione adottata) ossia una sequenza di passi



■ Rappresentazione

- Un modo di esprimere le informazioni rilevanti per risolvere il problema (dunque **NON TUTTE**)

■ Procedura

- Un “processo” a step (passi) da seguire per ottenere la soluzione

- In parole povere una **ricetta** ossia un **algoritmo**

Dunque una possibile *rappresentazione* della **scatola** in *centimetri* è:

- 1) Faccia A: Altezza, Larghezza (es. 220,70)
- 2) Faccia B: Altezza, Larghezza (es. 220,100)
- 3) Faccia C: Altezza, Larghezza (es. 70,110)

Mentre per la **porta** (*centimetri*):

- 1) Porta: Altezza, Larghezza (210, 80)

Esempio: 220,70,220,100,70,110,210,80

Sono solo **numeri!**

Il mondo esterno è **RAPPRESENTATO**
tramite numeri, memorizzati all'interno
del calcolatore

(1) *Verifica*: **se** ($\text{altezza}(A) < \text{altezza}(\text{porta})$) **e** ($\text{larghezza}(A) < \text{larghezza}(\text{porta})$)

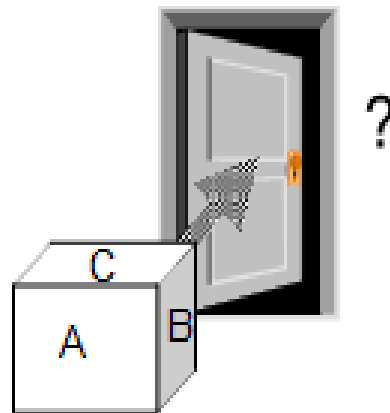
Se (1) è VERO **ok**; *altrimenti* verifica la faccia B (passo successivo);

(2) *Verifica*: **se** ($\text{altezza}(B) < \text{altezza}(\text{porta})$) **e** ($\text{larghezza}(B) < \text{larghezza}(\text{porta})$)

Se (2) è VERO **ok**; *altrimenti* verifica faccia C (passo successivo);

(3) *Verifica*: **se** ($\text{altezza}(C) < \text{altezza}(\text{porta})$) **e** ($\text{larghezza}(C) < \text{larghezza}(\text{porta})$)

Se (3) è VERO **ok**; *altrimenti* "**la scatola non passa**";



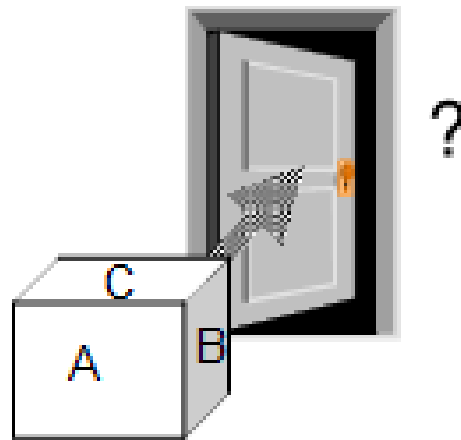
Abbiamo elaborato i dati giungendo ad una precisa informazione:

La scatola passa attraverso la porta

Oppure

La scatola NON passa attraverso la porta

Procedura "più generale" possibile



- La procedura appena vista vale per ogni porta e scatola?
- Siamo interessati a procedure il più possibile *general* e *valide* per il maggior numero possibile di casi, ossia ad algoritmi migliori

Algoritmo differente

(1) *Verifica*: **se** $\text{altezza}(A) < \text{altezza}(\text{porta})$ **e** $\text{larghezza}(A) < \text{larghezza}(\text{porta})$

Se (1) è VERO ok; *altrimenti* ruota faccia A e vai al passo 2;

(2) *Verifica*: **se** $\text{altezza}(A) < \text{altezza}(\text{porta})$ **e** $\text{larghezza}(A) < \text{larghezza}(\text{porta})$

Se (2) è VERO ok; *altrimenti* verifica faccia B;

(3..) *Etc.*

Quindi: questo algoritmo vale per un numero
MAGGIORE di scatole

Procedure "automatiche"

220,70,220,100,70,110,210,80

rappresentazione

+

(1)Verifica: **se** (altezza(A)<altezza(porta)) **e** (larghezza(A)<larghezza(porta))

Se (1) è VERO **ok**; altrimenti verifica la faccia B (passo successivo);

(2)Verifica: **se** (altezza(B)<altezza(porta)) **e** (larghezza(B)<larghezza(porta))

Se (2) è VERO **ok**; altrimenti verifica faccia C (passo successivo);

(3)Verifica: **se** (altezza(C)<altezza(porta)) **e** (larghezza(C)<larghezza(porta))

Se (3) è VERO **ok**; altrimenti "la scatola non passa";

Procedura
(algoritmo)

- “La scienza della rappresentazione dell’informazione ed elaborazione automatica”. Automatico?

Di macchina, meccanismo o dispositivo che, regolato opportunamente, è capace di compiere determinate operazioni o lavorazioni, per lo più ripetute in serie, senza il diretto intervento dell’uomo

Punto della situazione

- (X) Informazione
- (X) **Rappresentazione** (dell'informazione)
- (x) **Organizzazione** (dell'informazione)
- (X) Trattamento **automatico** (dell'informazione)
 - Algoritmi automatizzati

Algoritmi

- i passi costituenti devono essere "elementari", ovvero non ulteriormente scomponibili (*atomicità*);
- i passi costituenti devono essere interpretabili in modo diretto e univoco dall'*esecutore*, sia esso umano o artificiale (*non ambiguità*);
- l'algoritmo deve essere composto da un numero finito di passi e richiedere una quantità finita di dati in ingresso (*finitezza*);
- l'esecuzione deve avere termine dopo un tempo finito (*terminazione*);
- l'esecuzione deve portare a un risultato univoco (*effettività*).

Diagramma di flusso

- Linguaggio TRASVERSALE adatto a programmatori di ogni «estrazione linguistica»

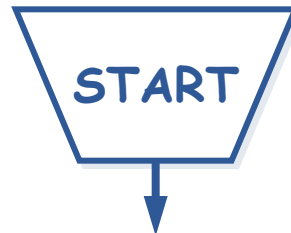
Un formalismo per scrivere "algoritmi"

X : 

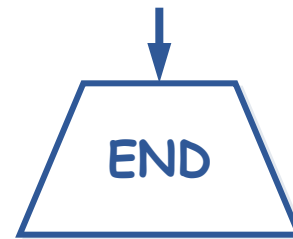
'contenitore' con
nome

$X \leftarrow expr$

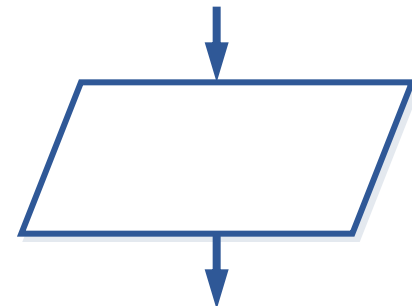
Mette il valore nel
"contenitore"



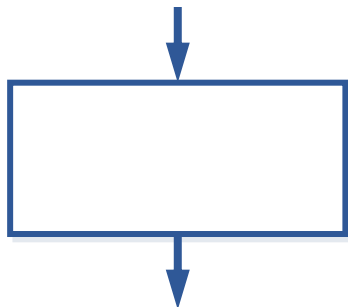
Inizia il processo
di calcolo



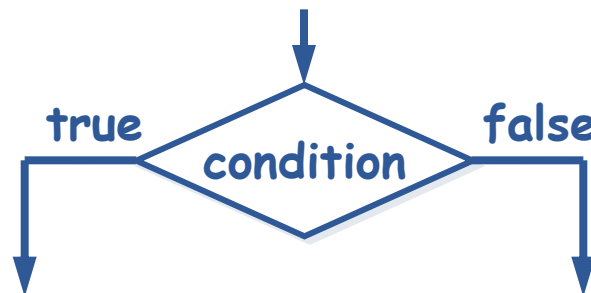
L'algoritmo
termina



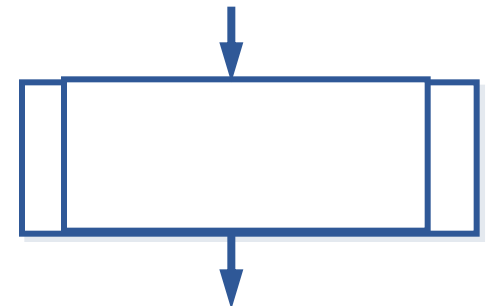
Ingresso /
Uscita



Un passo
dell'algoritmo



Selezione su
condizione



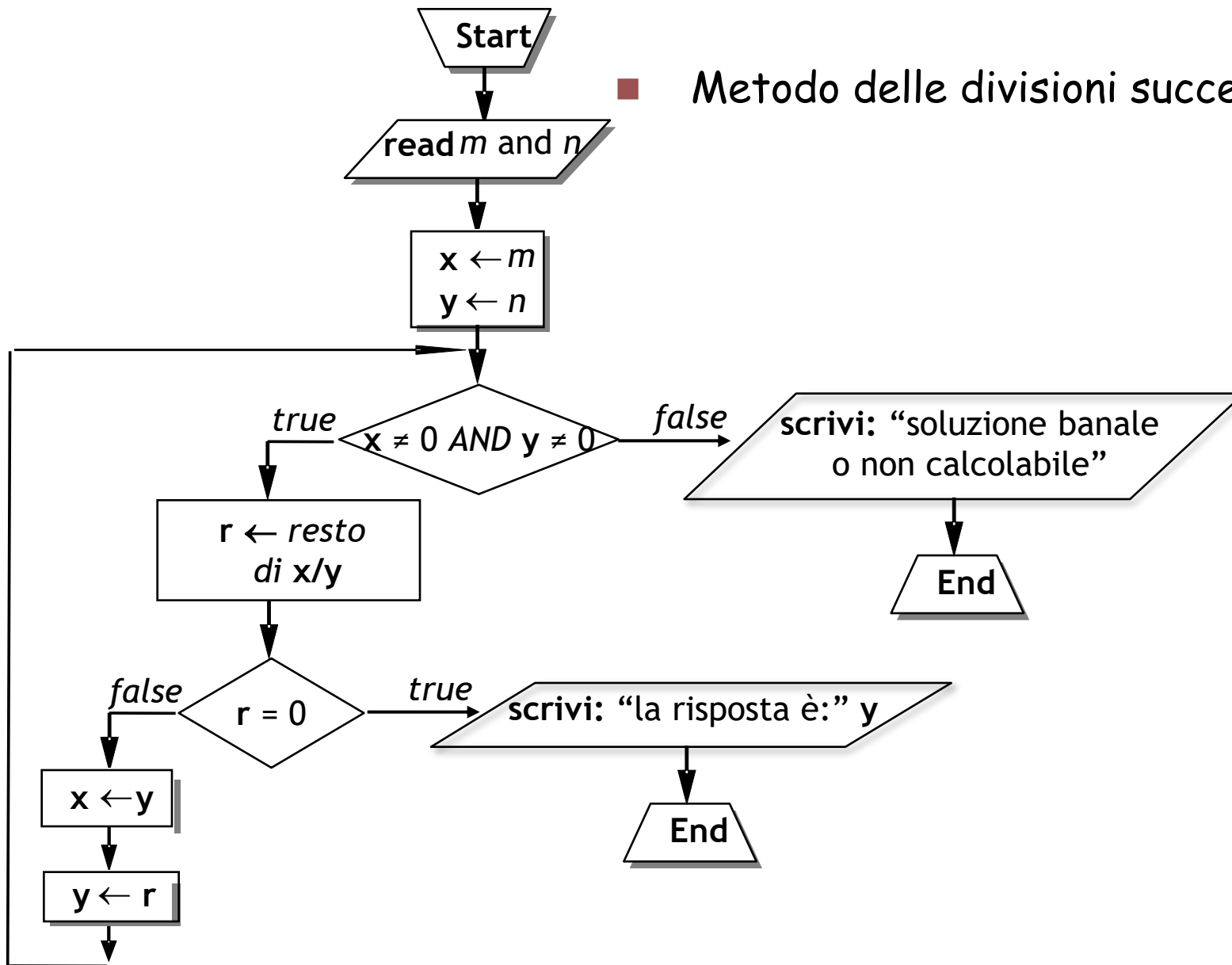
algoritmo "collaterale"

Dati due numeri interi “m” ed “n” con $m > n$:

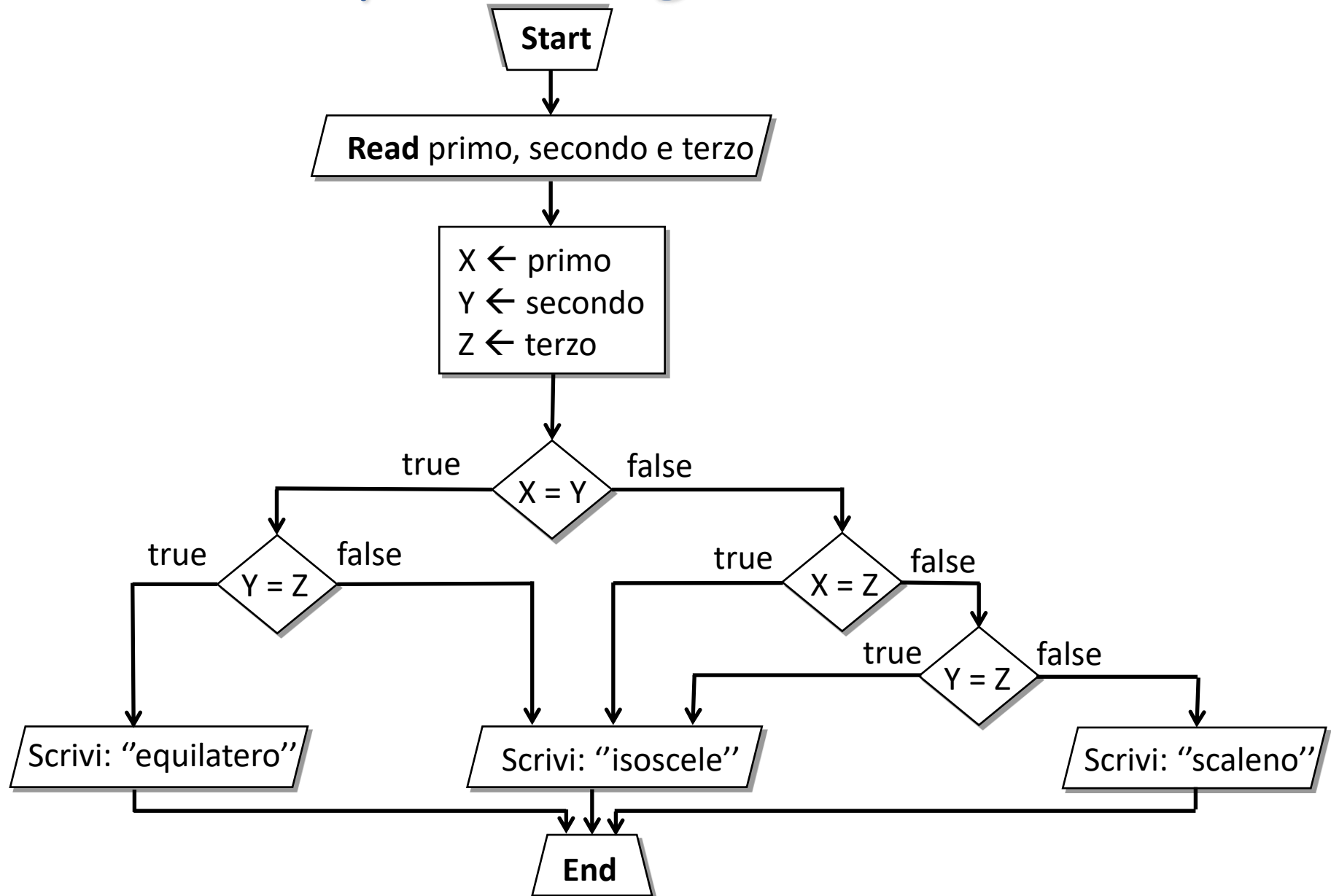
- 1) Si controlla se $n = 0$ (algoritmo moderno: ai tempi di Euclide non si conosceva lo zero);
- 2) Se $n = 0$, “m” è il risultato;
- 3) Se n diverso da 0 si calcola m/n e si assegna il resto alla variabile “r”;
- 4) Se “r” = 0 allora il risultato è “n”;
- 5) Se “r” è diverso da 0 allora si pone $m = n$, $n = r$ e si torna al passo 3.

Algoritmo espresso con diagramma di flusso

■ Metodo delle divisioni successive



Altro esempio di diagramma di flusso



Esercizio

Disegnare il diagramma di flusso dell'algoritmo della scatola che passa attraverso la porta

Conclusione

- Informazioni sul corso
- Materiale/siti
- Definizione di Informatica/algoritmo