Informatica

Terzo anno

Prof. A. Longheu

Come studiare???

- Anzitutto, qualche consiglio su come studiare
- Primo passo: capire quali fattori influenzano la propria capacità di imparare (ANALISI)

Come studiare???

- Secondo passo: STRATEGIA
- Qualche idea:
  - **organizzarsi**: pianificare i tempi
  - **autodisciplina**: imporsi tempi e ritmi
  - **orientarsi al risultato**: poche chiacchere, più azione
  - sperimenta e rifletti sulle tue **abitudini di studio**;
  - tieni il **controllo del tuo tempo libero**, ma quando ne hai rilassati e dimentica lo studio
  - **pianifica le pause** - trovando il modo migliore per farlo
  - **pianifica** la tua settimana guardando alle tue **abitudini** di studio preferite
  - prova a pianificare all'inizio dell'anno tenendo in considerazione le **perdite di tempo** per malanni, ecc.
  - Non prefiggerti obiettivi che non puoi raggiungere.

Come studiare???

- Ricerche effettuate affermano che l'84% degli studenti soffrono di problemi motivazionali durante i loro corsi. La **motivazione** è la chiave per il successo effettivo nello studio. La sua perdita può far cadere in un vortice verso il basso e può condurre ad una totale mancanza di successi.

Come studiare???

- Ma cosa significa “apprendere”???
- Qualche risposta:
  - Incamerare informazioni. Imparare è ricordare cose ed evitare la ripetizione di errori.
  - E' un processo di sviluppo attraverso il quale si cresce come persona... Quando hai realmente conquistato qualcosa senti un'enorme fiducia e vuoi trasmetterla agli altri.
  - E' la crescita della conoscenza e la comprensione del mondo. Ci rende capaci di un ulteriore progresso nella vita.
  - Apprendere ci rende autonomi nel pensiero e nell’azione; non devi dipendere dagli altri perché conosci e capisci

Come studiare???

- A lezione:
  - Prendere appunti: NON copiare, ma provare a riscrivere con parole proprie, aiuta a verificare subito se ho capito
  - Cercare di capire il filo logico del discorso: da dove si è partiti, cosa si sta facendo, dove si deve arrivare
  - Chiedere, chiedere, chiedere
  - Confrontarsi
- A casa:
  - Leggere: dai uno sguardo generale; interrogati sullo scopo della lettura; cerca i punti chiave e fai uno schema, anche grafico, di quello che stai leggendo.

Come studiare???

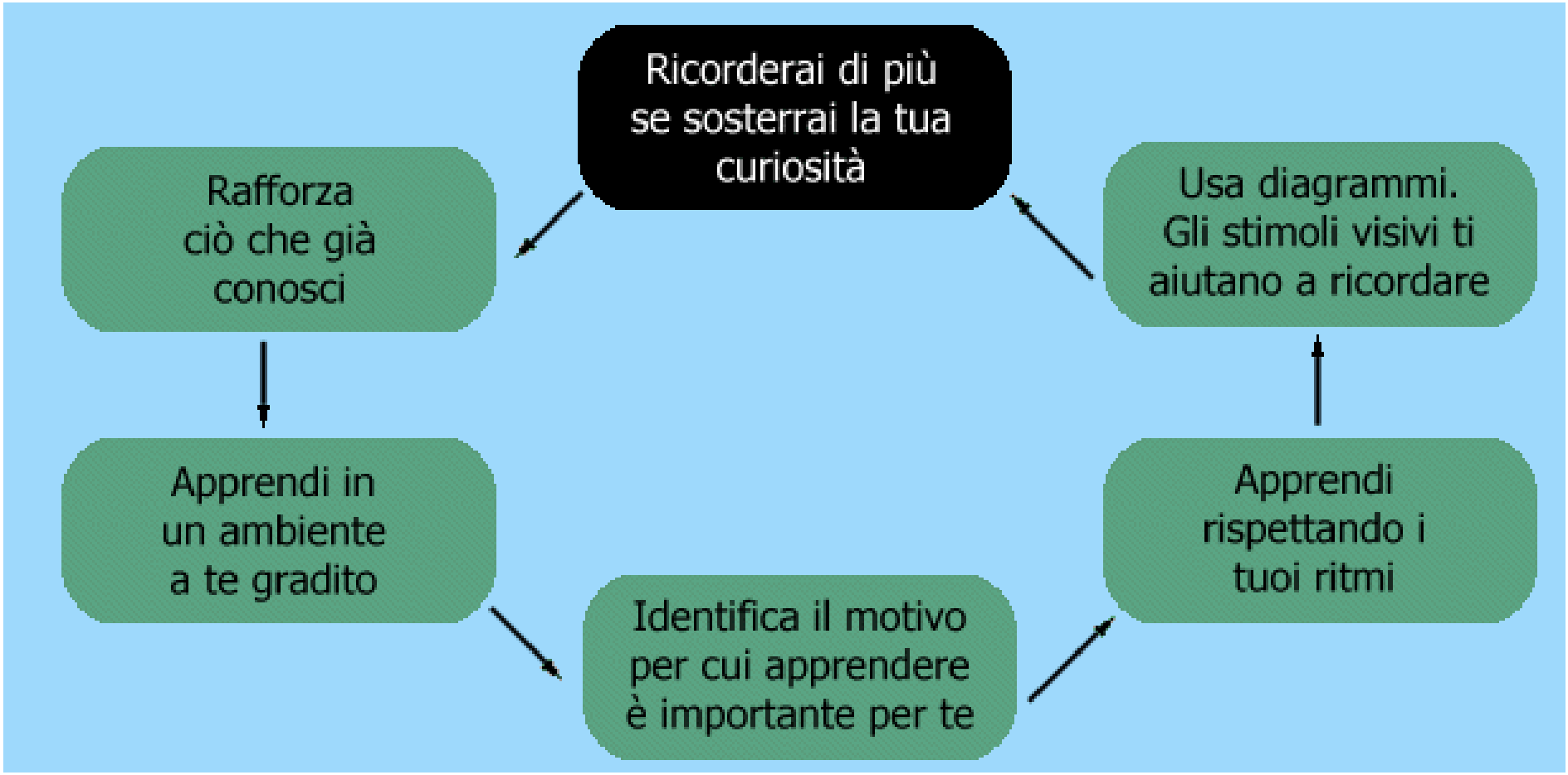
- Studiare per?????
  - Acquisire conoscenze (SAPERE)
  - Acquisire compentenze (SAPER FARE)
  - Ma soprattutto essere in grado di risolvere problemi di natura non abitudinaria.
- Studiare in gruppo:
  - ascolti idee di altre persone
  - esprimi le tue
  - fai pratica nel risolvere i problemi
  - ti rendi conto che non sei l'unico ad avere difficoltà
  - impari a comunicare

Come studiare???

- Durante un compito:
  - Leggi attentamente le **istruzioni** scritte dal docente
  - Leggi il foglio **dall'inizio alla fine**
  - sottolinea gli esercizi che pensi di poter svolgere più facilmente
  - Inizia proprio da questi ultimi, non badare all'ordine numerico
  - Se hai poco tempo e pensi di non riuscire a fare l'ultima domanda, fai un abbozzo, piuttosto che lasciarla in bianco.
  - Assicurati che la tua scrittura sia **leggibile**, il tuo italiano corretto, e ben chiaro quello che vuoi dire (sono più comuni di quanto pensi i problemi dovuti ad un compito illeggibile).
  - Non preoccuparti di chi ti sta accanto. Possono anche scrivere tanto, ma la **qualità** conta più della quantità.
  - Se ti lasci prendere dal **panico**, chiudi gli occhi, aspetta uno o due minuti respirando profondamente, finché non ti senti più calmo.
  - Se il tempo lo permette, dai un'occhiata finale al tuo lavoro, è impressionante il numero di errori, prettamente dettati dalla fretta e dallo stress, che potrai scoprire. Cerca di correggerli in tempo.

Come studiare???

- MEMORIA:



Storia dell'informatica

- Informatica = dal francese *information+automatique*, ovvero trattamento automatico delle informazioni
- →\storia_informatica:
  - Calcolatori meccanici:
    - abaco usato da romani, greci, cinesi (sec. I),
    - Pascal, Leibniz (~1600)
    - Telaio a schede perforate (1725)
    - Censimento USA con schede perforate, Hollerit IBM (1890)
  - Calcolatori elettromeccanici:
    - Colossus (Mark I e II), Alan Turing, Enigma (1945)
  - Calcolatori elettronici:
    - ENIAC (1946)

Storia dell'informatica

- Generazioni di calcolatori elettronici:
  - I: Valvole, 1946 (ENIAC)
  - II: Transistor, 1958
  - III: Circuiti integrati, 1964
  - IV: Microprocessori, anni 80
  - V: VLSI (very large scale integration), anni 90

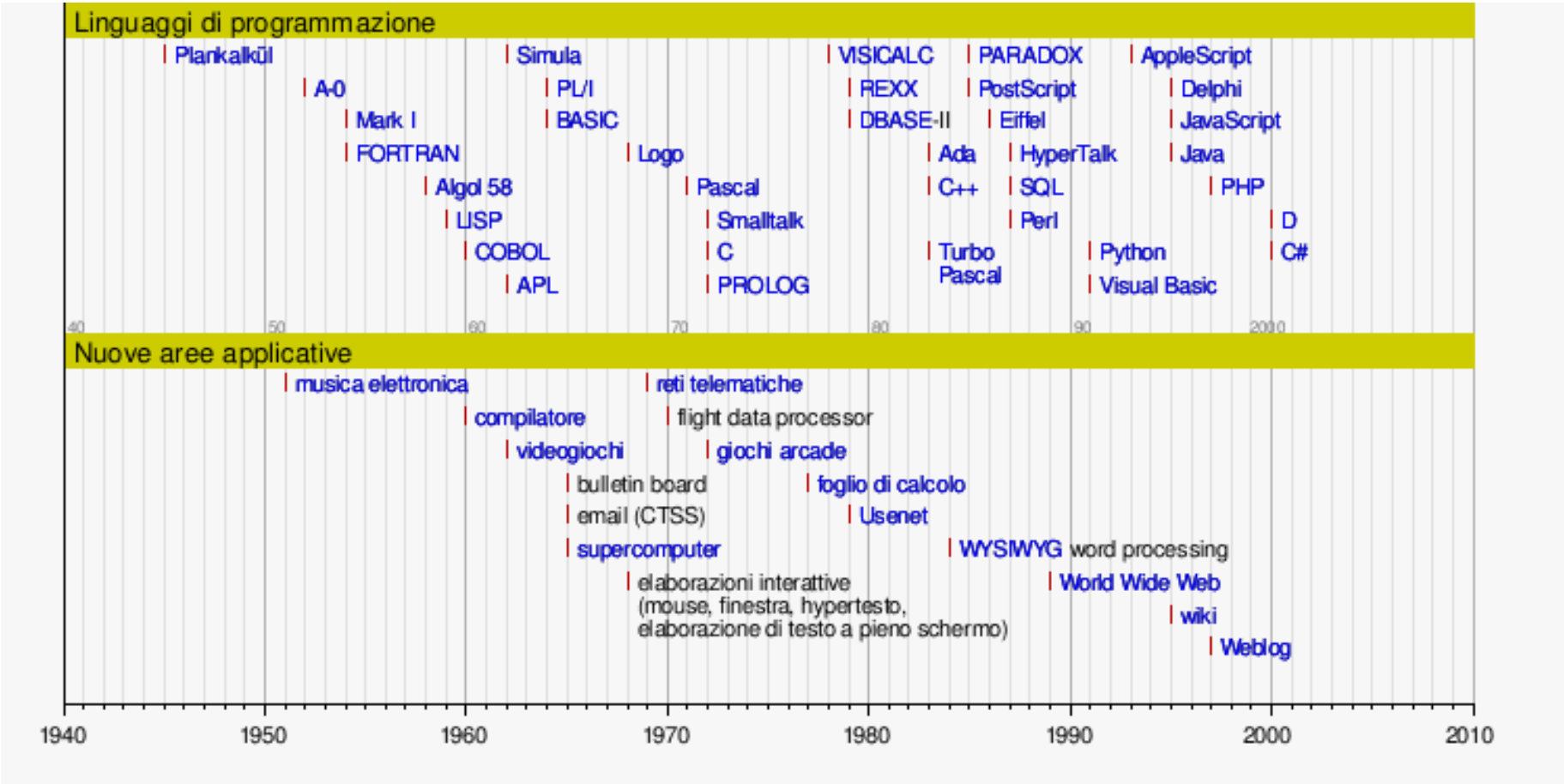
Storia dell'informatica

- Categorie di calcolatori in base alla dimensione:
  - Mainframe
  - Minicomputer
  - Microcomputer (gli odierni PC)

Storia dell'informatica

Storia dell'informatica

Storia dell'informatica



Scopo dell'informatica

- L'informatica si occupa del trattamento automatico delle informazioni
- Questo significa che la macchina deve leggerle, convertirle da un formato all'altro, analizzarle, trovare quelle di interesse, o anche crearne di nuove
- Tutte queste operazioni sono dei "problemi" che la macchina deve risolvere
- … o meglio, che l'umano deve risolvere per la macchina affinchè questa possa eseguire le operazioni e quindi essere utile per le nostre esigenze
- Il concetto di partenza quindi è quello di "problema"

Cosa e' un problema?

Hanno natura molto varia:

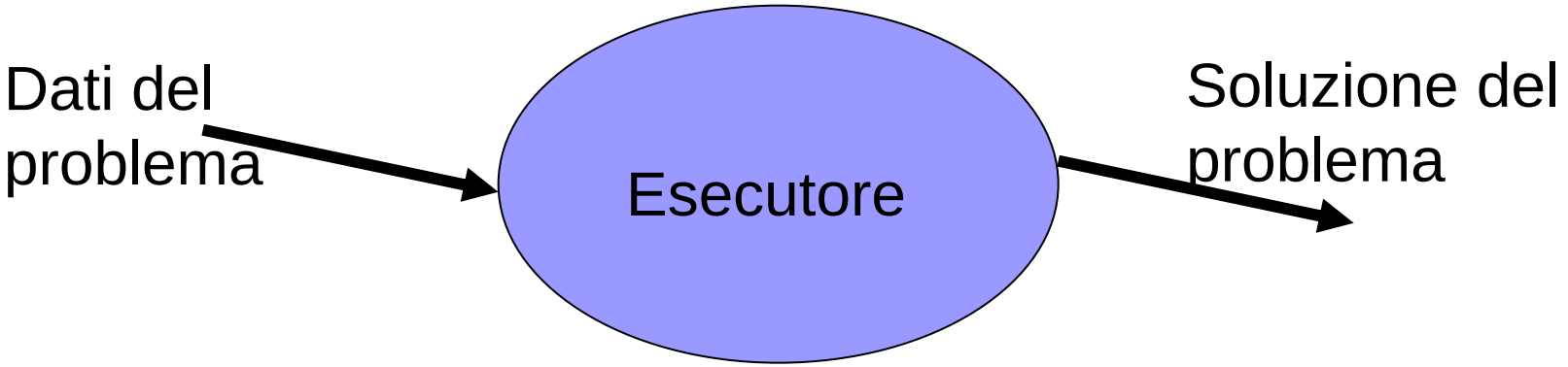
- Trovare il maggiore fra due numeri
- Dato un elenco di nomi e numeri di telefono, trovare il numero di una data persona
- Dati *a* e *b*, risolvere l'equazione $ax+b=0$
- Stabilire se una parola precede un'altra
- Identificare e prenotare aerei, treni, hotel...
- Ordinare una lista di elementi
- Creare, modificare e alterare suoni, canzoni, ...
- Analizzare e riconoscere immagini

Un problema è una domanda in attesa di una risposta

IL PUNTO

- La descrizione del problema **non indica** direttamente (in genere) un modo per ottenere il risultato voluto
- Differenza tra
  - *specifica di un **problema*** e
  - *specifica del **processo di risoluzione***

Esecutore: macchina astratta che è capace di eseguire alcune azioni



Problema: Risolvere un problema significa ricercare ed esprimere un elenco di istruzioni che interpretate da un esecutore conducono da deteminate informazioni iniziali ad altre informazioni finali soddisfacenti un criterio di verifica

Risoluzione di un problema

- E’ quel processo che
  - dato un *problema*
  - individuato un opportuno *metodo risolutivo*
- trasforma
  - i ***dati iniziali***
  - nei corrispondenti ***risultati finali***

Alcune domande fondamentali

- ***Quali istruzioni*** *esegue un elaboratore?*
- ***Quali problemi*** *può risolvere un elaboratore?*
- ***Esistono problemi che un elaboratore non può risolvere****?*
- *Che ruolo ha il linguaggio di programmazione?*

Algoritmo - Esecutore

- Bisogna **prima fissare un insieme di "mosse elementari"** possibili per l'elaboratore
- **Ma... è corretto** impostare la soluzione a partire da tali mosse elementari ?
  - **SI**, per risolvere il problema ***in modo efficiente***
  - **NO**, se la macchina di partenza ha mosse di livello ***troppo basso***
- **Occorre risalire a un più adeguato *livello di astrazione***

Cosa serve?

- Metodi per la rappresentazione dell'informazione (DATI)
- Metodi per la descrizione delle modalità con cui ottenere una soluzione (ISTRUZIONI)
- Caratterizzazione del risolutore dei problemi (LINGUAGGIO)

Problema ben formulato

- non è a priori evidente che non esistono soluzioni
- il criterio di verifica delle soluzioni è univoco e si sa come applicarlo
- l'insieme dei dati iniziali è completo

Problema ben formulato

Si definisce effettiva per un esecutore la soluzione di un problema allorchè:

- l'esecutore è in grado di interpretare i dati di ingresso
- l'esecutore è in grado di interpretare la descrizione di tale soluzione, e quindi di associare ad essa le azioni che deve compuere per eseguirla
- l'esecutore è in grado di compiere tali azioni, completando l'esecuzione in un tempo finito

Algoritmo

- Un algoritmo è il procedimento risolutivo di un problema
- E' l'insieme di regole che, eseguite ordinatamente, permettono di ottenere i risultati del problema a partire dai dati a disposizione

Non tutte le regole formano un algoritmo

Algoritmo

- Def. ISO: Un algoritmo è un insieme finito e ordinato di regole formalizzate destinato a fornire la soluzione di un problema, in un numero finito di passi
- Affinchè un insieme di regole possano essere considerare un algoritmo devono rispettare alcune proprietà

Proprietà di un algoritmo

- **Non ambiguità**: ogni azione elementare indicata nell'algoritmo deve produrre un risultato ben definito, per tutti i possibili valori di ingresso
- **Eseguibilità**: ogni passo dell'algoritmo deve essere effettivamente eseguibile, in un tempo finito, dall'esecutore considerato
- **Determinismo**: nell'esecuzione dell'algoritmo, per ogni passo compituto deve essere determinato uno e un solo passo successivo.

Proprietà di un algoritmo

- **Finitezza**: l'algoritmo deve essere costruito da passi discreti e la sua lunghezza deve essere finita
- **Terminazione**: l'esecuzione dell'algoritmo deve, prima o poi, terminare.

Esempi di algoritmi

Calcolare la soluzione di $ax + b = 0$

1. leggere i valori di a e di b
2. calcolare -b
3. dividere -b per a e assegnare il risultato ad x
4. scrivere x

Esempi di algoritmi

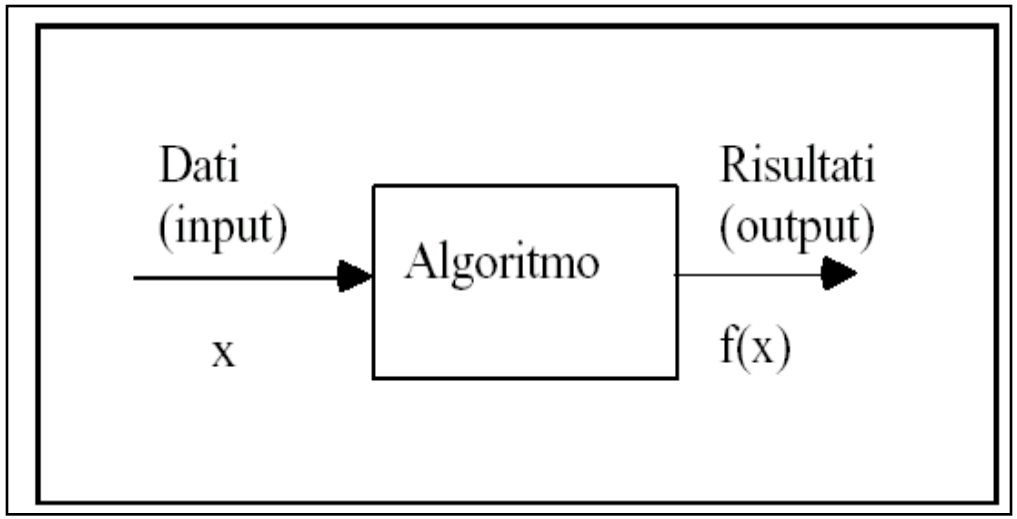
Problema del trasporto della capra, del lupo e del cavolo da una sponda ad un'altra del fiume

1. Porta la capra sull'altra sponda
2. Torna indietro
3. Porta il cavolo sull'altra sponda
4. Porta la capra indietro
5. Porta il lupo sull'altra sponda
6. Torna indietro
7. Porta la capra sull'altra sponda

ALGORITMI EQUIVALENTI

*Due algoritmi si dicono **equivalenti** quando:*

- *hanno lo stesso dominio di ingresso;*
- *hanno lo stesso dominio di uscita;*
- *in corrispondenza degli stessi valori nel dominio di ingresso* producono gli stessi valori *nel dominio di uscita*



ALGORITMI EQUIVALENTI

Due algoritmi equivalenti:

- *forniscono lo **stesso risultato***
- *ma possono avere **diversa efficienza***
- *e possono essere **profondamente diversi !***

***Esempio:** moltiplicare tra loro due numeri*

Algoritmo 1

Somme successive:

12x12 = 12+12+…+12=144

Algoritmo 2
"somma e shift":

```
 12x
 12=
 --
 24
12=
---
144
```

CI SONO COSE CHE UN CALCOLATORE NON PUÓ FARE?

- Sì. Ci sono problemi non calcolabili da nessun modello di calcolo reale o astratto

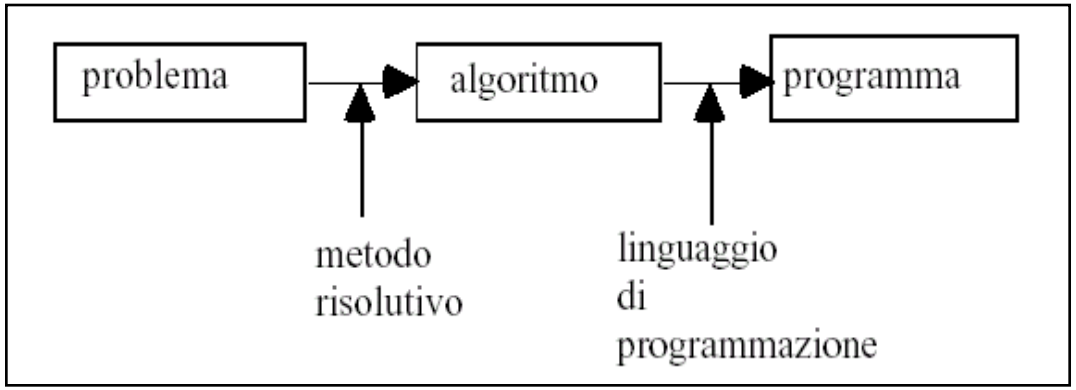
- Esempio: data una funzione f : N$\rightarrow$ N, stabilire se f(x) è costante per ogni valore di x

Processo di risoluzione

- Analisi del problema ed identificazione di una soluzione
- Formalizzazione della soluzione e definizione dell'algoritmo risolutivo
- Scrittura di un programma in un linguaggio di programmazione
- Traduzione del programma in un linguaggio compreso dal calcolatore
- Esecuzione del programma

ALGORITMO E PROGRAMMA

- *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*



ALCUNI CONCETTI CHIAVE

ALGORITMO
Sequenza finita di mosse che risolve in un tempo finito una classe di problemi.

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.*

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