

Introduzione al corso

Introduzione all'Informatica - corso E
Docente: Ing. Irina Trubitsyna

Obiettivi del corso

- ◆ Concetti di base sull'informatica e sulla gestione delle informazioni
- ◆ Struttura e funzionamento dei personal computer
- ◆ Architettura delle reti informatiche
- ◆ Principali funzioni dei sistemi operativi
- ◆ Uso dei principali strumenti di produttività individuale (gestione di testi, fogli elettronici)
- ◆ Uso di strumenti per la navigazione, la ricerca e la comunicazione su Internet

Programma (teoria)

- ◆ Definizione di Informatica. Hardware e software. Algoritmi e programmi.
- ◆ Rappresentazione delle informazioni. Codifica di numeri, caratteri, immagini e dati multimediali.
- ◆ Architettura dei calcolatori. La macchina di Von Neumann. Unità centrale di elaborazione (CPU). Dispositivi di memoria. Funzionalità dei sistemi operativi.
- ◆ Reti di calcolatori. Tassonomia delle reti per estensione e topologia. Scambio di informazioni: protocolli e mezzi di trasmissione. Nozioni sui meccanismi di commutazione. La rete Internet. Cenni sul protocollo TCP/IP. Il modello client/server. Il Web e le principali applicazioni di Internet. Indirizzi, nomi simbolici e DNS. Cenni sull'HTML ed i documenti ipertestuali.

Programma (esercitazioni)

- ◆ Uso del sistema operativo Windows. Utilità di sistema.
- ◆ **Software di produttività individuale:**
 - MS Word
 - MS Excel
- ◆ Strumenti per la navigazione sul Web (uso e configurazione di un browser).
- ◆ Ricerca di informazioni su Web (uso dei motori di ricerca).
- ◆ Posta elettronica e gestione dei messaggi.

Materiale didattico

◆ Libri di testo:

■ Teoria

D. Sciuto, G. Buonanno, L. Mari, “Introduzione ai sistemi informatici”, terza edizione, McGraw-Hill, 2005.

■ Esercitazioni

La guida McGraw-Hill alla patente europea del computer, McGraw-Hill, 2002

◆ Lucidi ed altro materiale:

<http://wwwinfo.deis.unical.it/~irina/>

Modalità di esame

◆ Prova scritta/pratica (durata 1 ora):

- 1 esercizio di Word (10 punti)
- 1 esercizio di Excel (10 punti)
- 1 esercizio di Teoria (2 domande, 10 punti)

◆ Condizioni per il superamento della prova scritta:

- Conseguire almeno 5 punti per ciascun esercizio.

◆ Prova orale facoltativa:

- La prova orale è obbligatoria solo se il voto della prova scritta è pari a 15, 16 o 17

Esonero dall'esame

- ◆ Gli studenti in possesso della Patente Europea del Computer (ECDL) possono richiedere l'esonero dall'esame
- ◆ Il modulo per l'esonero è disponibile sul sito della Facoltà:

<http://www.ingegneria.unical.it/webingegneria/ecdl>

Altre informazioni

◆ *Docente:*

- **Irina Trubitsyna**, DEIS, cubo 41C, VI° piano
e-mail: irina@deis.unical.it
riceve giovedì, dalle 12:00 alle 13:00

◆ *Tutor:*

- **Luciano Caroprese**
e-mail: caroprese@deis.unical.it
- **Antonella Dimasi**
e-mail: dimasi@exeura.it

Orario

gruppo 1 : MARIGLIANO Gianmarco - MURACE Bruno

gruppo 2 : MURANO Fabrizio - PATE Roberto

	LUNEDI'	MARTEDI'	MERCOLEDI'	GIOVEDI'	VENERDI'
8.30		Lab. gruppo 1		Lezione	
9.30		" "		(Aula 32B1)	
10.30		Lab. gruppo 2			
11.30		" "			
12.30					
14.30					
15.30					
16.30					
17.30					
18.30					

Lezione 1:

Concetti introduttivi

Docente: Irina Trubitsyna

Informatica
Algoritmo
Programma

Informazione e comunicazione

◆ Mondo fisico, materiale

◆ Mondo dell'Informazione

- Leggi scientifiche
- Sinfonie
- ...

} Il prodotto dell'interesse dell'uomo a conoscere il mondo che lo circonda e a comunicare le sue conoscenze, le sue esperienze ed i suoi stati d'animo.

Informazione e comunicazione

◆ Informazione

- Notizia, dato o elemento che consente di avere conoscenza più o meno esatta di fatti e situazioni

◆ Messaggio

- Tutto ciò che porta informazione

◆ Comunicazione

- Scambio di informazione, mediante **messaggi**

Cos'è l'informatica?

◆ Esistono varie definizioni:

- Scienza dell'informazione
- Informazione + automatica: gli strumenti e le tecniche utilizzati nel trattamento automatico delle informazioni
- Scienza dei calcolatori ("Computer Science")
- *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 [Académie Française]*

Cos'è l'informatica?

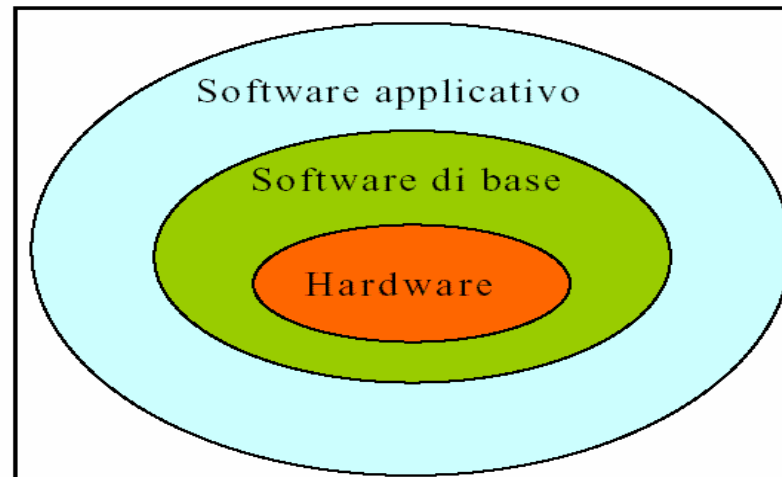
- ◆ **Informatica = Scienza della rappresentazione e dell'elaborazione dell'informazione**
 - Studia le caratteristiche dell'informazione ed i modi di usarla, immagazzinarla, elaborarla e trasportarla in modo automatico
- ◆ **L'informatica ha due anime:**
 - **tecnologica**: studia i calcolatori elettronici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

Elaboratore elettronico

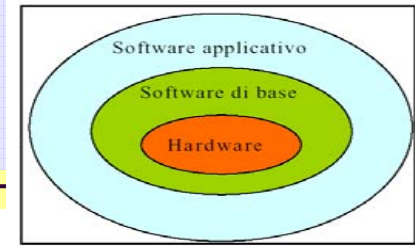
- ◆ Elaboratore elettronico (o “computer” o “calcolatore”)
 - è uno strumento **programmabile** per rappresentare, memorizzare ed elaborare informazioni
- ◆ La prima decomposizione di un calcolatore è relativa alle seguenti macro-componenti
 - **Hardware:** la struttura fisica del calcolatore costituita da dispositivi di varia natura: elettronici, elettromagnetici, elettromeccanici, ottici ...
 - **Software:** l'insieme dei programmi che consentono all'hardware di svolgere dei compiti utili all'utente

Classificazione del software

- ◆ **Software di Base** (es. il Sistema Operativo)
 - permette una più semplice interazione con le componenti hardware (memorie, periferiche, ...)
- ◆ **Software Applicativo**

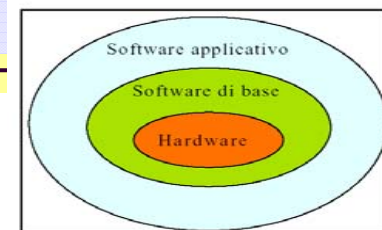


Hardware, software



- ◆ L'hardware è l'unica macchina reale, mentre i vari strati software corrispondono a macchine virtuali
 - le operazioni (**istruzioni**) che l'hardware sa eseguire direttamente rappresentano le frasi del **linguaggio macchina** del calcolatore
 - le istruzioni del linguaggio macchina sono molto semplici e 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



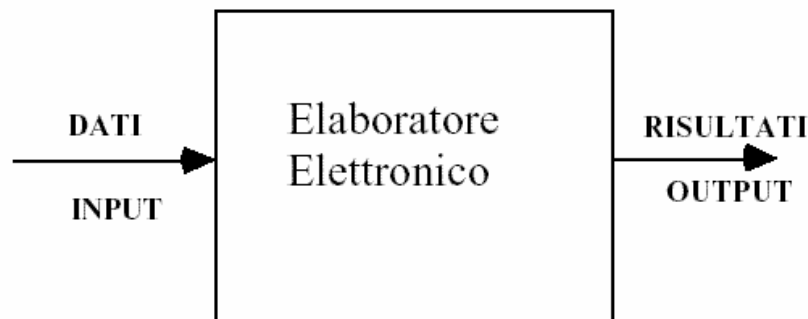
◆ Macchine virtuali

- Semplificano la comunicazione fra uomo e hardware
- Le diverse macchine ed 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

Elaboratore elettronico



◆ Alcune domande fondamentali:

- Quali istruzioni esegue un elaboratore?
- Quali problemi può risolvere un elaboratore?
- Esistono problemi che un elaboratore non può risolvere?

◆ 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 e complessità molto varia, es.:

- 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 di un'organizzazione (private e pubbliche)
- Supportare operazioni di commercio elettronico

I problemi

◆ Descrizione del problema

- La descrizione del problema non indica direttamente (in genere) un modo per risolverlo

specifica di un problema
≠
specifica del processo di risoluzione

◆ Risoluzione di un problema

- Comprensione
- Modellazione
- Individuazione di un opportuno metodo risolutivo
(**algoritmo di risoluzione**)

Risoluzione di un problema

L'obiettivo fondamentale

Descrizione di un problema



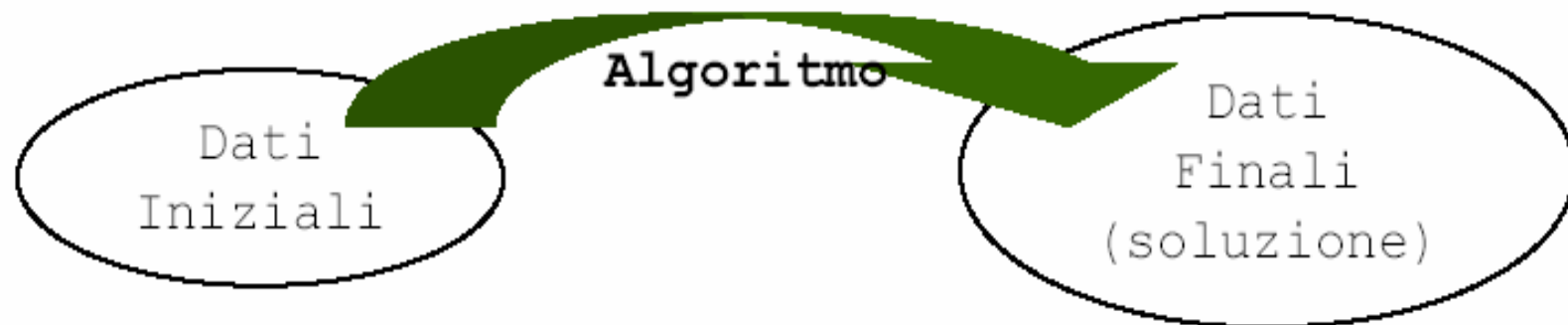
Progetto

Individuazione di un ALGORITMO

Algoritmo

- ◆ **Algoritmo** = sequenza finita di passi, elementari e non ambigui, che risolve un problema in un tempo finito
- ◆ Esempi di “algoritmi”:
 - Istruzioni di montaggio
 - Preparazione del caffè
 - Prelievo bancomat
 - Ricetta di cucina
 - Calcolo del *massimo comun divisore* tra due interi

Algoritmo



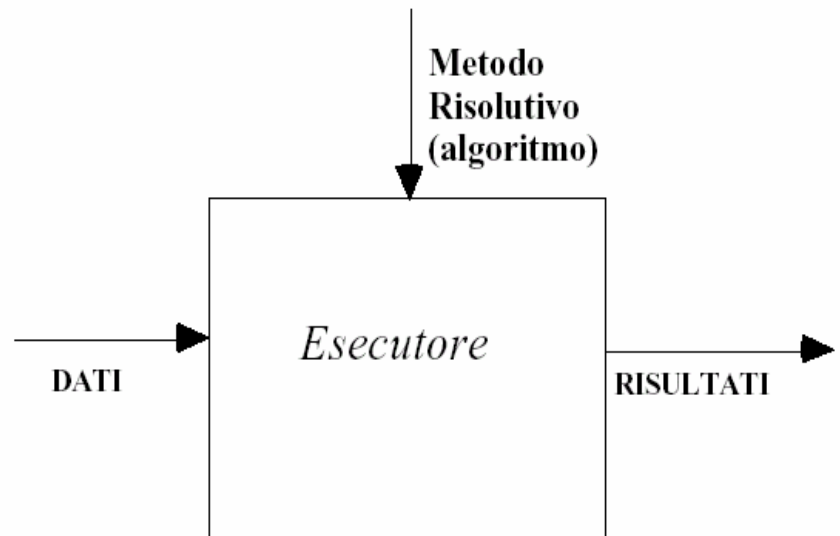
Si definisce *algoritmo* una *sequenza di azioni* che trasformi i dati iniziali in un numero finito di passi, elementari e non ambigui, per giungere al risultato finale.

Questa sequenza di azioni è valida per un insieme di dati iniziali ben definito e può essere eseguita da un opportuno esecutore.

Algoritmo: 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** del problema

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



Proprietà degli algoritmi

◆ Proprietà fondamentali

- **Generalità**: applicabile a ogni insieme di dati di ingresso appartenente al dominio di definizione del problema
- **Non-ambiguità**: ogni azione deve essere univocamente interpretabile dall'esecutore (persona o “macchina”)
 - costituito da operazioni appartenenti ad un determinato insieme di operazioni fondamentali
- **Eseguibilità**: ogni azione deve essere eseguibile in un tempo finito da parte dell'esecutore dell'algoritmo
- **Finitezza**: per ogni insieme di dati di ingresso, il numero totale di azioni da eseguire deve essere finito

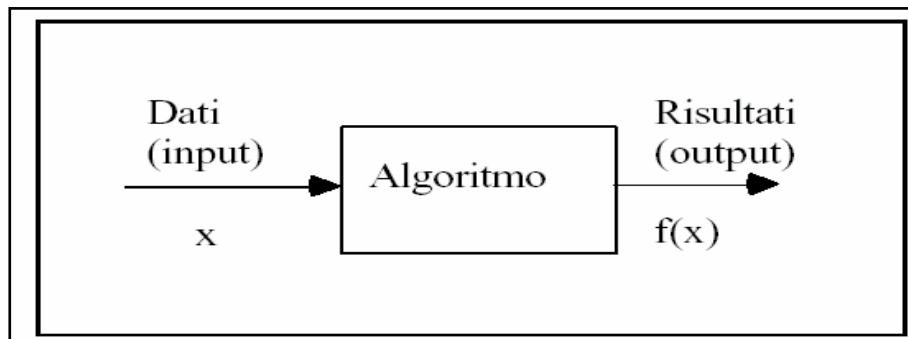
◆ Proprietà desiderabile

- **Efficienza**: deve risolvere il problema utilizzando al meglio le risorse a disposizione

Algoritmi equivalenti

In generale un algoritmo può essere visto come una **funzione**

- da un dominio di ingresso (**input**)
- ad un dominio di uscita (**output**)



Due algoritmi si dicono **equivalenti** quando:

- 1) hanno stesso dominio di ingresso e stesso dominio di uscita;
- 2) in corrispondenza degli stessi valori nel dominio di ingresso producono gli stessi valori nel dominio di uscita

Problemi non risolvibili

Non ammettono algoritmi di risoluzione con nessun modello di calcolo reale o astratto

Esempio:

- predire il valore delle azioni FIAT nel 2010
- predire se una certa squadra vincerà il campionato di calcio

Rappresentazione degli algoritmi

1. Linguaggio naturale

Linguaggi informali

2. Diagrammi di flusso

Linguaggi semi-formali

3. Pseudo-codice

4. Linguaggio di programmazione

Linguaggi formali

ESEMPIO: calcolo del MCD

Problema:

- Dati due interi M ed N (input)
- calcolare il Massimo Comun Divisore (MCD) fra M e N

*Algoritmo 1** (descritto in linguaggio naturale):

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

Presuppone l'esistenza di un esecutore in grado di interpretarlo ed eseguirlo!

Calcolo del MCD: un altro algoritmo

Metodo di Euclide

$$\text{MCD}(M, N) = \begin{cases} M \text{ (oppure } N) & \text{se } M=N \\ \text{MCD}(M-N, N) & \text{se } M>N \\ \text{MCD}(M, N-M) & \text{se } M<N \end{cases}$$

Algoritmo 2

1. Finché $M \neq N$
 - se $M > N$, sostituisci a M il valore $M-N$
 - altrimenti sostituisci a N il valore $N-M$
2. Il Massimo Comun Divisore è M (o N), cioè il valore finale ottenuto quando M e N diventano uguali

I simboli M e N sono due **variabili**



Variabili (cenni)

- ◆ Rappresentano dei “contenitori” per dati
- ◆ Ogni variabile possiede
 - nome
 - tipo
 - dominio dei valori che può assumere (es., numeri interi, caratteri,...)
 - valore
 - caratterizza lo stato della variabile, che può cambiare durante l'esecuzione
- ◆ Esempio:
 - $N = 2$ (il valore iniziale di N è 2)
 - $N = N + 3$ (il valore corrente di N è 5)

Variabili: esempio

Calcoliamo il MCD di $M = 24$ e $N = 14$.

1. $M=24, N=14$
 $24 > 14 \rightarrow M = 24 - 14 = 10$
2. $M=10, N=14$
 $10 < 14 \rightarrow N = 14 - 10 = 4$
3. $M=10, N=4$
 $10 > 4 \rightarrow M = 10 - 4 = 6$
4. $M=6, N=4$
 $6 > 4 \rightarrow M = 6 - 4 = 2$
5. $M=2, N=4$
 $2 < 4 \rightarrow N = 4 - 2 = 2$
6. $M=2, N=2$
 $2 = 2 \rightarrow$ “il MCD di 24 e 14 è 2”

(Algoritmo)

1. Finché $M \neq N$

- se $M > N$, sostituisci a M il valore $M - N$
- altrimenti sostituisci a N il valore $N - M$

2. Il MCD è $M (=N)$

Calcolo del MCD (3)

Algoritmo n° 3

Dati due interi M e N ($M \geq N$)

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

Osservazione

I tre algoritmi visti per il calcolo del MCD sono equivalenti, ma differiscono per efficienza

Calcolo del MCD (3): applicazione

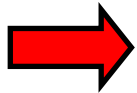
Calcoliamo il MCD di $M = 24$ e $N = 14$.

1. **$M=24, N=14$**
 $24/14 = 1, R=10 \rightarrow M=N=14, N=R=10$
2. **$M=14, N=10$**
 $14/10 = 1, R=4 \rightarrow M=N=10, N=R=4$
3. **$M=10, N=4$**
 $10/4 = 2, R=2 \rightarrow M=N=4, N=R=2$
4. **$M=4, N=2$**
 $4/2 = 2, R=0 \rightarrow$ “il MCD di 24 e 14 è 2”

(Rappresentazione degli algoritmi)

1. Linguaggio naturale

Linguaggi informali



Diagrammi di flusso

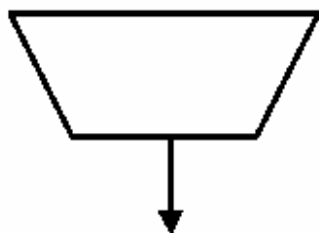
Linguaggi semi-formali

3. Pseudo-codice

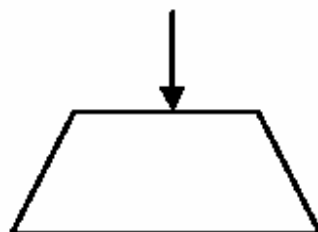
4. Linguaggio di
programmazione

Linguaggi formali

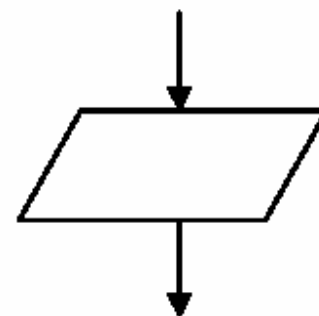
Diagrammi di flusso



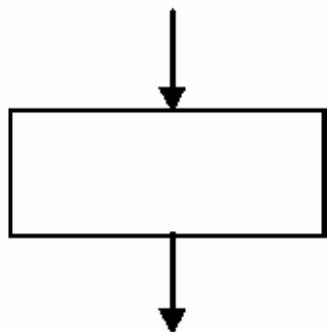
Inizio



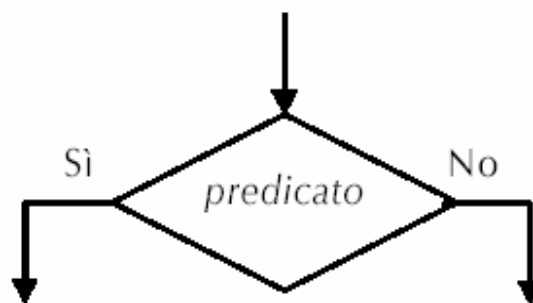
Fine



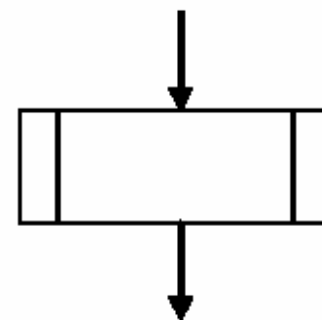
Operazioni
di ingresso/uscita



Elaborazione

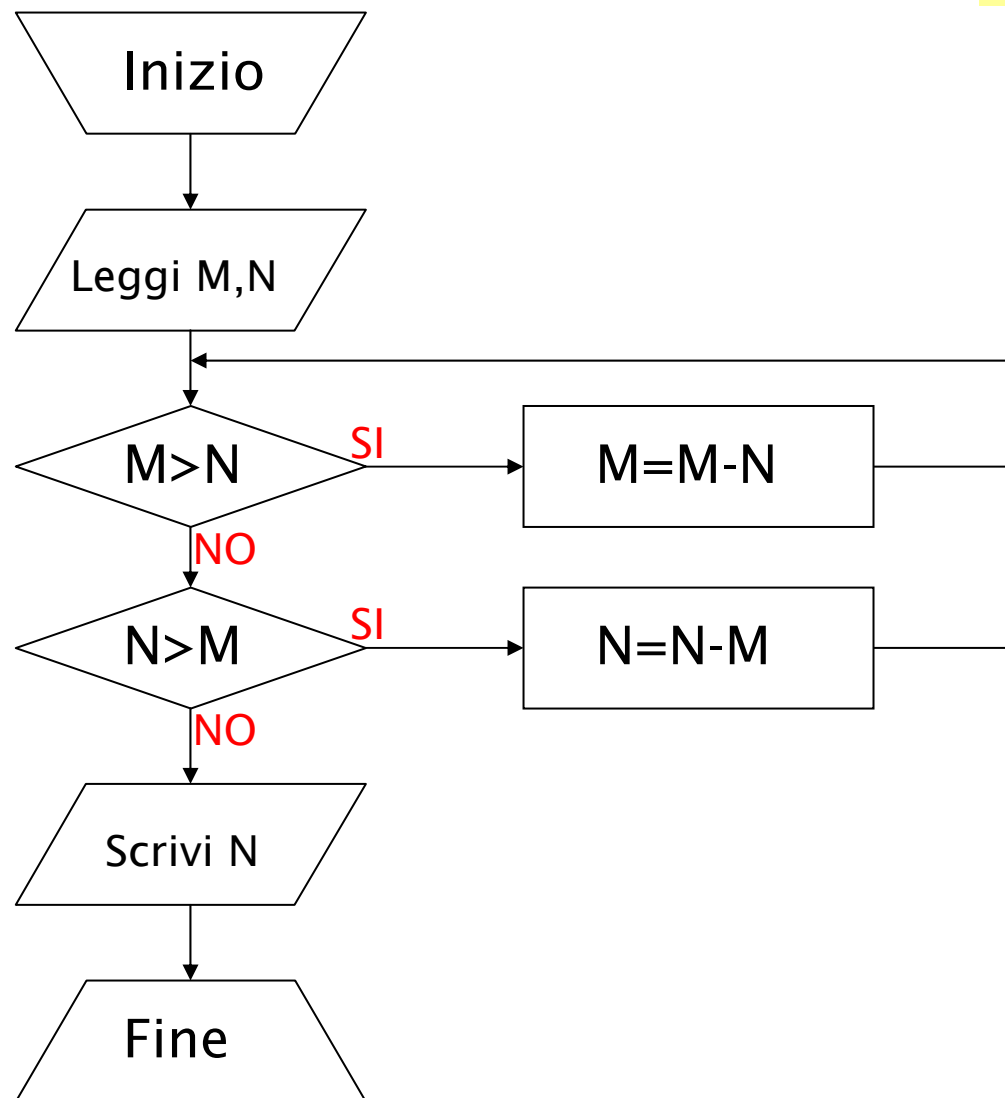


Selezione a due vie

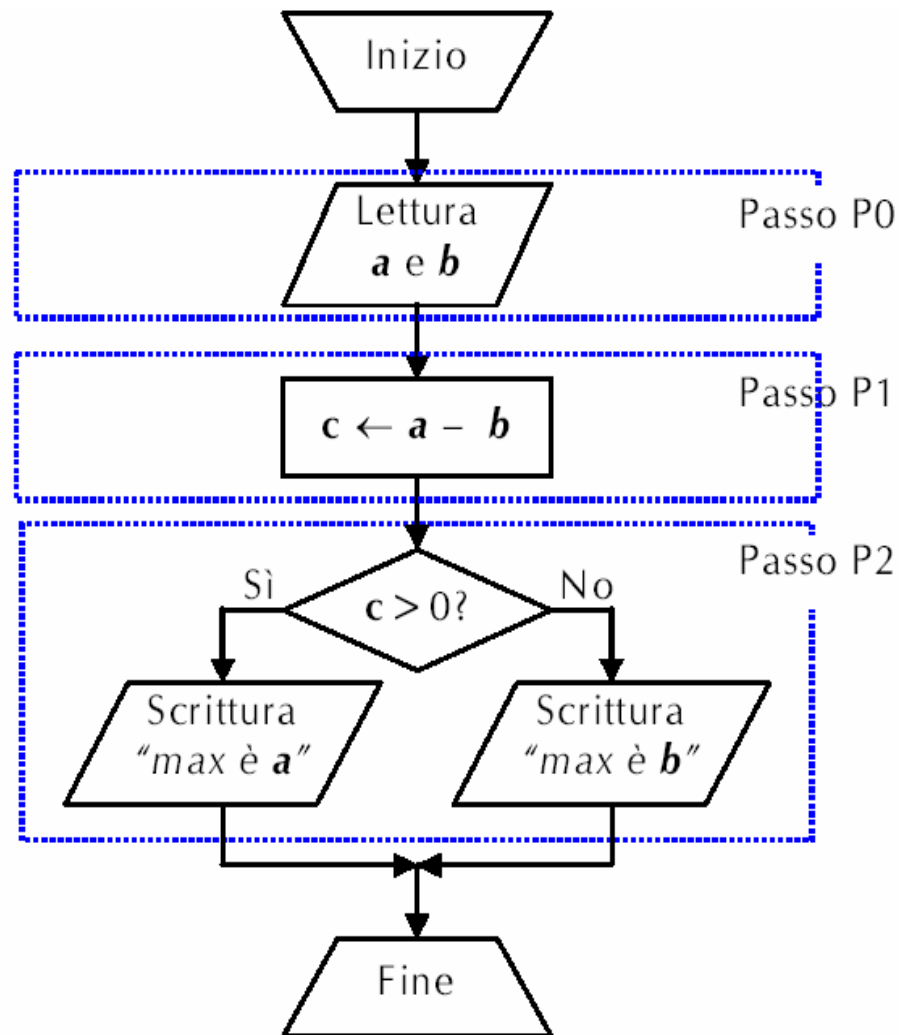


Sottoprogramma

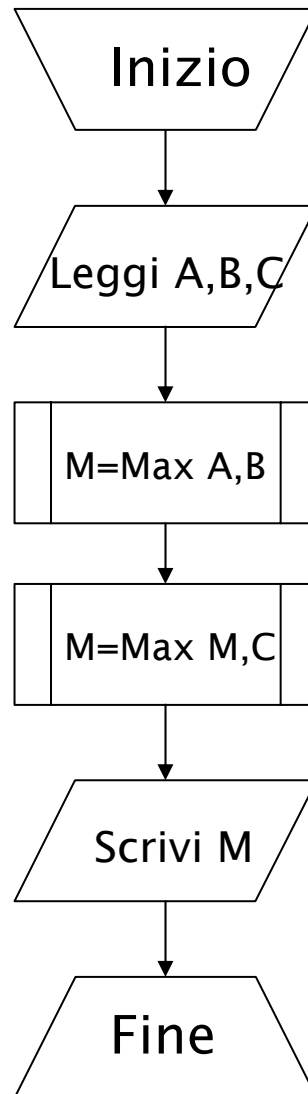
Esempio: MCD



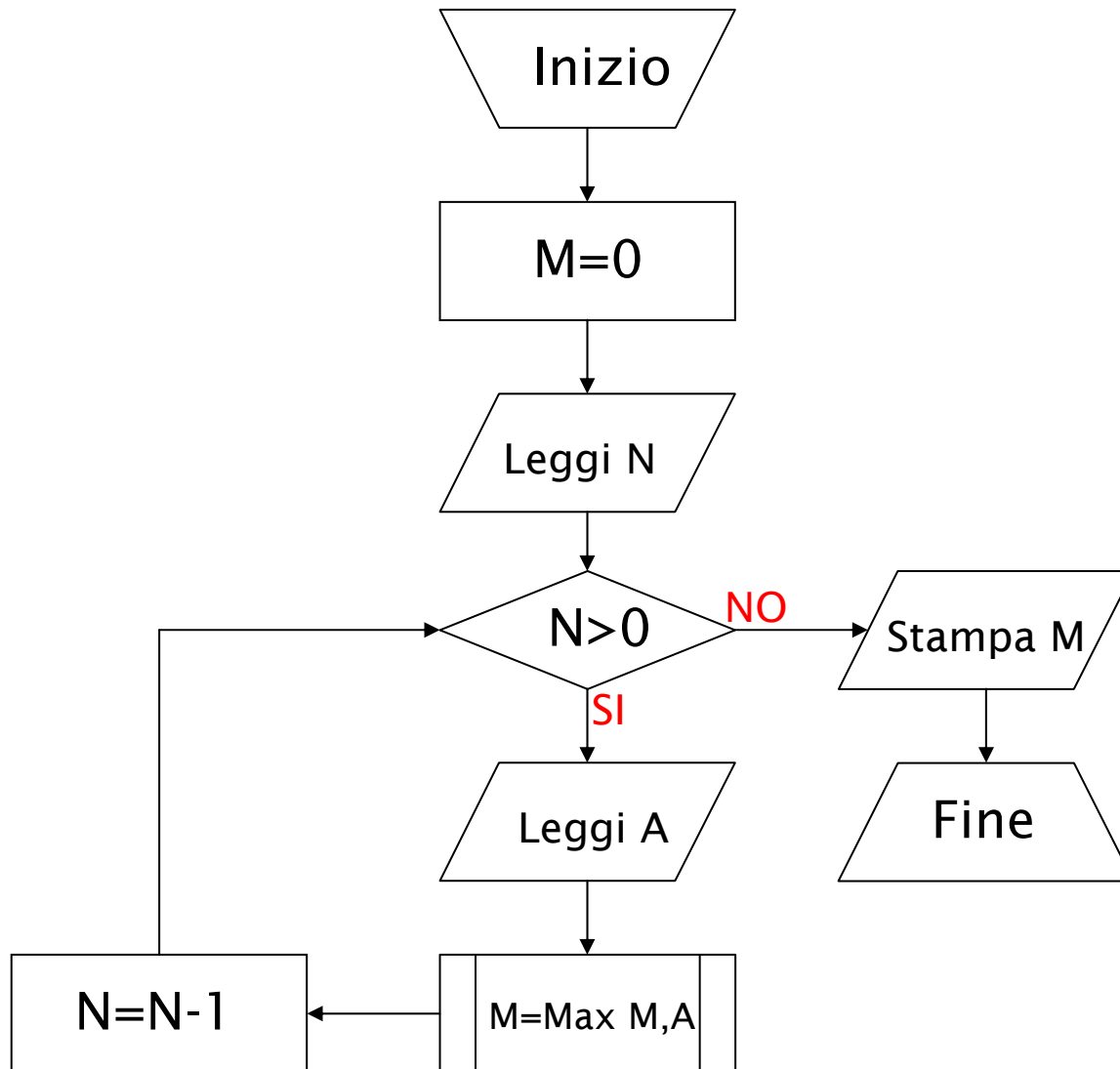
Esempio: massimo tra due numeri



Esempio: massimo tra tre numeri



Esempio: massimo tra N numeri positivi



(Rappresentazione degli algoritmi)

1. Linguaggio naturale

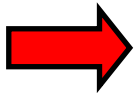
Linguaggi informali

2. Diagrammi di flusso

Linguaggi semi-formali

3. Pseudo-codice

Linguaggi formali

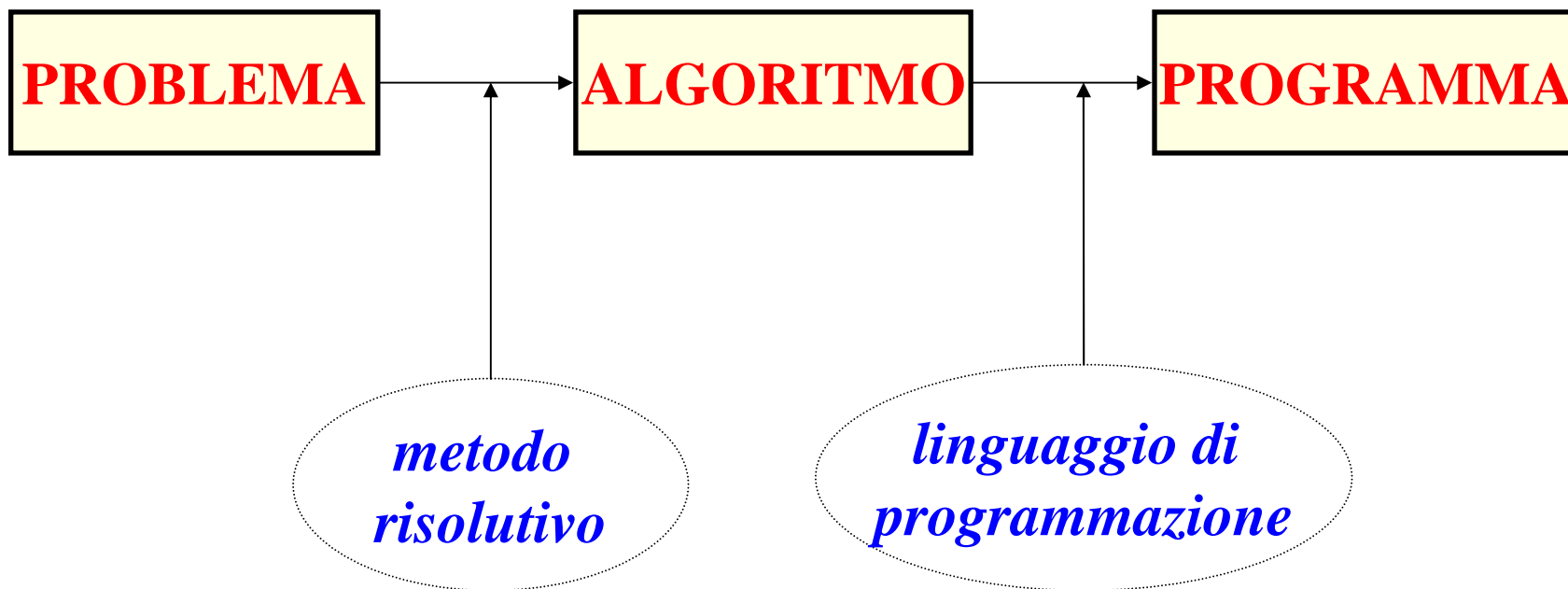


**Linguaggi di
programmazione**

Risoluzione di problemi con il calcolatore

- ◆ Un calcolatore (elaboratore) è una macchina (reale o virtuale) 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**
 - **linguaggio macchina** nel caso dell'hardware
- ◆ Un **programma** è la formulazione testuale di un algoritmo in un linguaggio di programmazione
 - in accordo alla sintassi e alla semantica del linguaggio di programmazione scelto

Algoritmi e programmi



Elementi tipici di un linguaggio di programmazione

◆ Operazioni elementari

- Operazioni aritmetiche e assegnamento di valori a singole variabili

Es. $C = (A + B)$ $(C \leftarrow A + B)$

- Condizioni sul valore di singole variabili

Es. Se $A > B$ allora ...

- Lettura e scrittura di variabili

Es. **Leggi** A oppure **Stampa** B

◆ Istruzioni di controllo del flusso

- Specificano l'ordine di esecuzione delle varie operazioni di un programma

Strutture di controllo: SEQUENZA

- ◆ Le istruzioni devono semplicemente essere eseguite nell'**ordine** in cui sono presentate:
 - 1- solleva il ricevitore
 - 2- componi il numero
 - 3- ...
- ◆ Una sequenza di istruzioni può essere **raggruppata** in modo da diventare una nuova macro-istruzione:

INIZIO

solleva il ricevitore
componi il numero
...

FINE

Strutture di controllo: CONDIZIONE

- ◆ Le istruzioni da eseguire sono determinate dalla valutazione di una data **condizione**

- ◆ Esempio:

SE il numero è libero

ALLORA

attendi la risposta

conduci la conversazione

deponi il ricevitore

ALTRIMENTI

deponi il ricevitore

Strutture di controllo: ITERAZIONE

- ◆ Le istruzioni devono essere eseguite ripetutamente fino a che non si verifica una determinata condizione
- ◆ Esempio:

RIPETI

componi il numero

FINO a che la linea è libera

Esempio: Calcolo della potenza (algoritmo)

◆ **Problema:**

Dati due interi **a** e **n** calcolare la potenza **a^n**

◆ **Algoritmo (in pseudo-codice):**

1. inizializza le **variabili** $K = n$, $Ris = 1$
2. fino a che $K > 0$
 - 2.1 calcola $Ris \cdot a$ e memorizzalo in Ris
 - 2.2 decrementa K

Correttezza: al termine $Ris = a^n$

Linguaggi ad alto livello

- ◆ Conviene 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**
 - Alto livello di astrazione: le istruzioni corrispondono ad operazioni più complesse
 - esempi: Pascal, Basic, C, C++, Java
 - E' necessario tradurre il programma nel linguaggio macchina mediante opportuni programmi (**interprete** o **compilatore**)

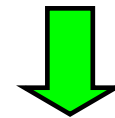
Esempio: calcolo della potenza (programma)

Programma (in pseudo-Pascal):

```
PROGRAM potenza;  
INTEGER Ris,N,A;  
BEGIN  
    READ(N);  
    READ(A);  
    Ris=1;  
    WHILE (N>0) DO  
        BEGIN  
            Ris=Ris*A;  
            N=N-1;  
        END;  
    PRINT(Ris);  
END.
```

L'esecutore deve:

1. leggere i valori iniziali dei parametri (N e A) dall'**input** (es., tastiera)
2. stampare il risultato (valore finale di Ris) sull'**output** (es, video)



Il programma (in linguaggio ad alto livello) deve essere tradotto nel **linguaggio macchina** del calcolatore