



**Dipartimento di Informatica e Sistemistica
Antonio Ruberti**

“Sapienza” Università di Roma

Introduzione

Corso di Fondamenti di Informatica

Laurea in Ingegneria Informatica

(Canale di Ingegneria delle Reti e dei Sistemi Informatici - Polo di Rieti)

Anno Accademico 2007/2008

Prof. Paolo Romano

Si ringrazia il Prof. Alberto Finzi per aver reso
disponibile il materiale didattico sul quale si basano queste slides

Obiettivo del Corso

Acquisizione di:

- ***Nozioni di informatica di base:***

nozioni sul funzionamento di un elaboratore elettronico e sulla rappresentazione delle informazioni.

- ***Programmazione:***

principi fondamentali, le nozioni e i metodi della programmazione (imperativa) introdotti utilizzando il linguaggio di programmazione C.

Materiale Didattico

- **Informatica di base:**

Dispense distribuite durante il corso. Disponibili anche sulla [pagina base del corso](http://www.dis.uniroma1.it/~romanop) (<http://www.dis.uniroma1.it/~romanop>)

- **Programmazione in C:**

H. Deitel, P. Deitel. *C Corso completo di Programmazione*. Apogeo.

Modalità di Esame

Modalità esame:

L'esame si compone di due prove:

- prova pratica: programmazione in C.
- prova teorica: domande di informatica di base.

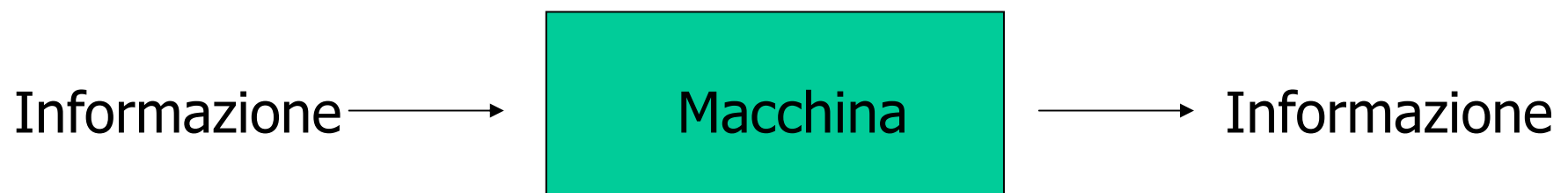
Introduzione

Introduzione

Informatica: Informazione + Automatica

Disciplina che studia:
Elaborazione Automatica delle Informazioni

Automatica = Eseguita da una Macchina

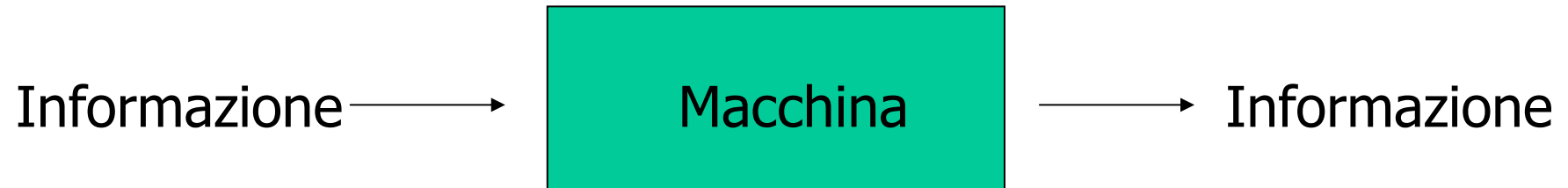


Elaborazione = Calcolo

Elaboratore = Macchina Calcolatrice

(Scienza dell'Informazione / Computer Science)

Introduzione

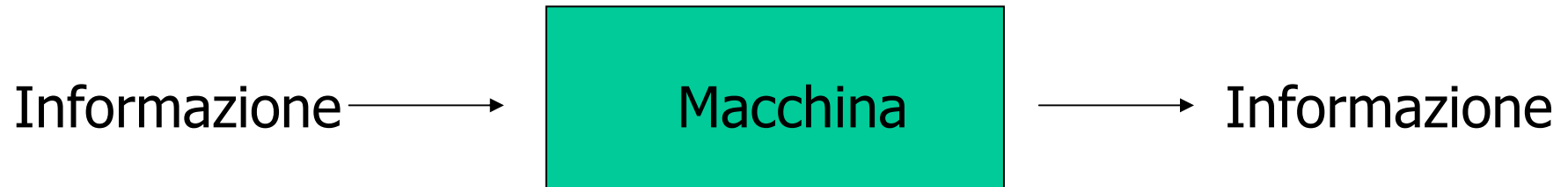


Tre aspetti: Informazione, Macchina, Calcolo

Problematiche connesse:

- Rappresentazione dell'informazione.
- Funzionamento della macchina calcolatrice.
- Procedure e metodi per il calcolo automatico.

Esempio Macchina Calcolatrice



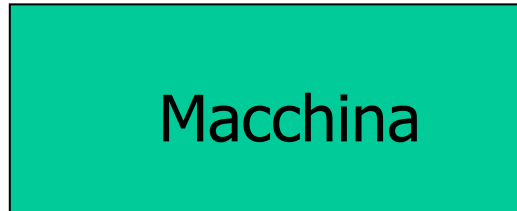
Esempio Calcolatrice: (Blaise Pascal 1642)



Calcolo $(10 * 5 + 30) \dots$
controllo manuale della sequenza (o flusso) operazioni

Esempio Macchina Calcolatrice

Informazione →



→ Informazione

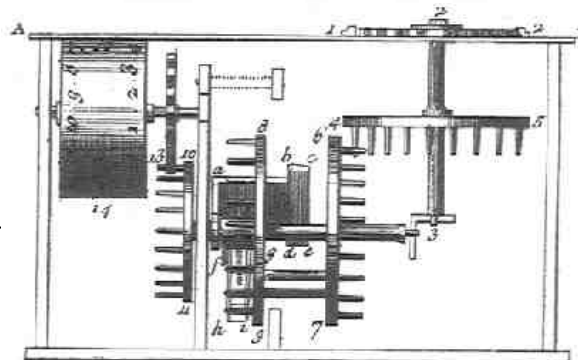
(Blaise Pascal 1642)

numeri →

operazione →



→ numero

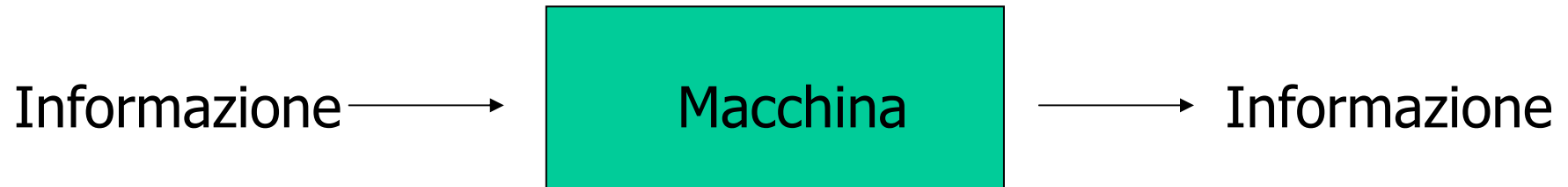


Rapp. informazione ingresso

calcolo

Rapp. informazione uscita

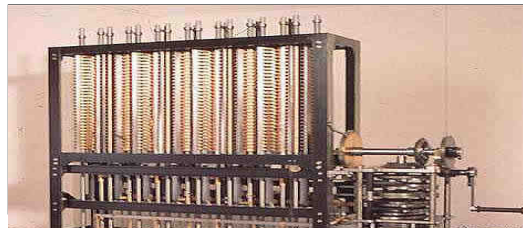
Macchina Calcolatrice Programmabile



Macchina Calcolatrice Programmabile:

Informazione
ingresso →

Procedura di
calcolo →



([Charles Babbage 1833](#))

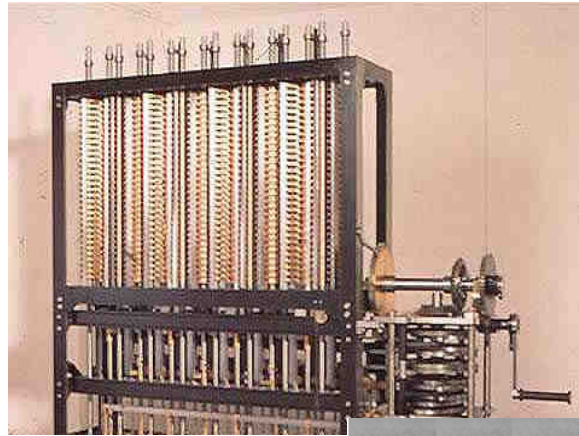
→ Informazione
uscita

controllo automatico della sequenza di operazioni (flusso)
(Tra le informazioni in ingresso anche la procedura di calcolo)

Il calcolatore ([computer](#)/ordinateur) esegue istruzioni nella modalità prevista dal metodo di calcolo.

Macchina Programmabile

THE ANALYTICAL ENGINE



macchina alle differenze

E' considerata la prima programmatrice: descrive come calcolare i numeri di Bernoulli con la macchina analitica.

Ada Byron



Ada, Countess of Lovelace, at age 27 (sketch by A. E. Chalon)



Charles Babbage

Storia Calcolatore

1642 - **Blaise Pascal**, prima macchina calcolatrice: [PASCALINE](#).

1674 - **W. Leibnitz** usa il **sistema binario**: [CALCOLATORE A SCATTI DI LEIBNITZ](#)

1816 - **Charles Babbage**, progetta [MACCHINA ALLE DIFFERENZE](#).

1884 - **H. Hollerith** costruisce una macchina per archiviare i dati per il censimento di New York (fonda **IBM**)

1904 - **Fleming** inventa la **valvola termoionica**

1945 - [PRIMA GENERAZIONE DI COMPUTERS \(Valvole\)](#).

1946 - Nasce l' **ENIAC** funzionante a valvole termoioniche e quindi velocissimo rispetto ai calcolatori realizzati fino a quel momento.

1947 - Invenzione del **transistore**.

1950- **John Von Neumann**, viene realizzato il primo calcolatore, l'**EDVAC**, in grado di memorizzare internamente un programma.

1951 - Nasce l' **UNIVAC**, il primo calcolatore ad essere prodotto su scala industriale.

1955 - [SECONDA GENERAZIONE DI COMPUTERS \(Transistori\)](#).

1959 - Viene inventato il **circuito integrato**.

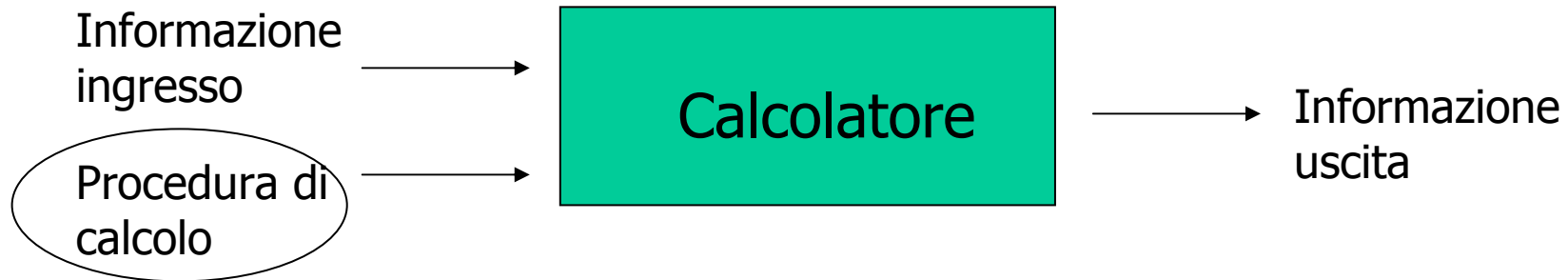
1968 - [TERZA GENERAZIONE DI COMPUTERS \(Circuiti integrati\)](#).

1971 - Viene inventato il **microprocessore**.

1974 - [QUARTA GENERAZIONE DI COMPUTERS \(Microprocessori\)](#).

1980 - Nasce il **personal computer**.

Algoritmo



Metodo **effettivo** per il calcolo: **ALGORITMO** (dall'algebrista Al-Khuwarizmi)

Algoritmo: *metodo risolutivo (insieme ordinato di istruzioni) che trasforma un insieme di dati iniziali in un insieme di risultati finali.*

Proprietà Algoritmo:

- a. Non ambiguità:* interpretazione univoca delle operazioni.
- b. Eseguitività:* ogni istruzione deve essere eseguibile.
- c. Finitezza:* i passi indicati dall'algoritmo sono finiti.

Esempio della ricetta

PENNE ALICI E PACHINO:

Ingredienti (Dati)

400 grammi di alici fresche, 300 grammi di pomodorini pachino, 400 grammi di pasta formato penne rigate, 2 rametti di prezzemolo
1 spicchio d'aglio, 1 peperoncino fresco o secco, 1 cucchiaino di sale fino, 2 cucchiari di sale grosso, 5 cucchiari di olio di oliva

Metodo di preparazione (Calcolo)

In un tegame ampio soffriggi l'aglio, l'olio ed il peperoncino. Unisci i filetti di alici fresche precedentemente lavate, eviscerate e private della lisca centrale. Fai cuocere a fuoco moderato per 10 minuti con un coperchio. Unisci i pomodorini già lavati e tagliati ciascuno in quattro parti. Aggiungi il sale fino e fai cuocere altri 10 minuti, sempre con il coperchio. Unisci un rametto di prezzemolo già lavato, e sminuzzato. Lascia cuocere pochi minuti senza coperchio. Nel frattempo fai cuocere la pasta in acqua bollente e salata con il sale grosso. Versa il condimento sulla pasta scolata, mescola ed aggiungi l'altro rametto di prezzemolo sempre sminuzzato. Servi in tavola ben caldo.

Algoritmi e Programmi

L'algoritmo fornisce la specifica astratta di una metodologia di calcolo.

L'algoritmo è dato in ingresso alla macchina. Per descrivere alla macchina l'algoritmo occorre un *linguaggio di programmazione*.

Un *linguaggio di programmazione* (imperativo) è un linguaggio per rappresentare alla macchina le istruzioni di un algoritmo e l'ordine in cui devono essere eseguite.

Un *programma* è la specifica di un algoritmo in un linguaggio di programmazione.

Programmazione

Con la prog. vogliamo risolvere classi di problemi fornendo un metodo risolutivo traducibile in un linguaggio di programmazione

Problema → Algoritmo → Programma

Problema: dati in ingresso/uscita

Algoritmo: metodo di calcolo (soluzione astratta)

Programma: codifica algoritmo nel LDP (implementazione)

Fase Problema/Algoritmo:

- analisi struttura del problema;
- specificazione della soluzione come operazioni sui dati.

Fase Algoritmo/Programma:

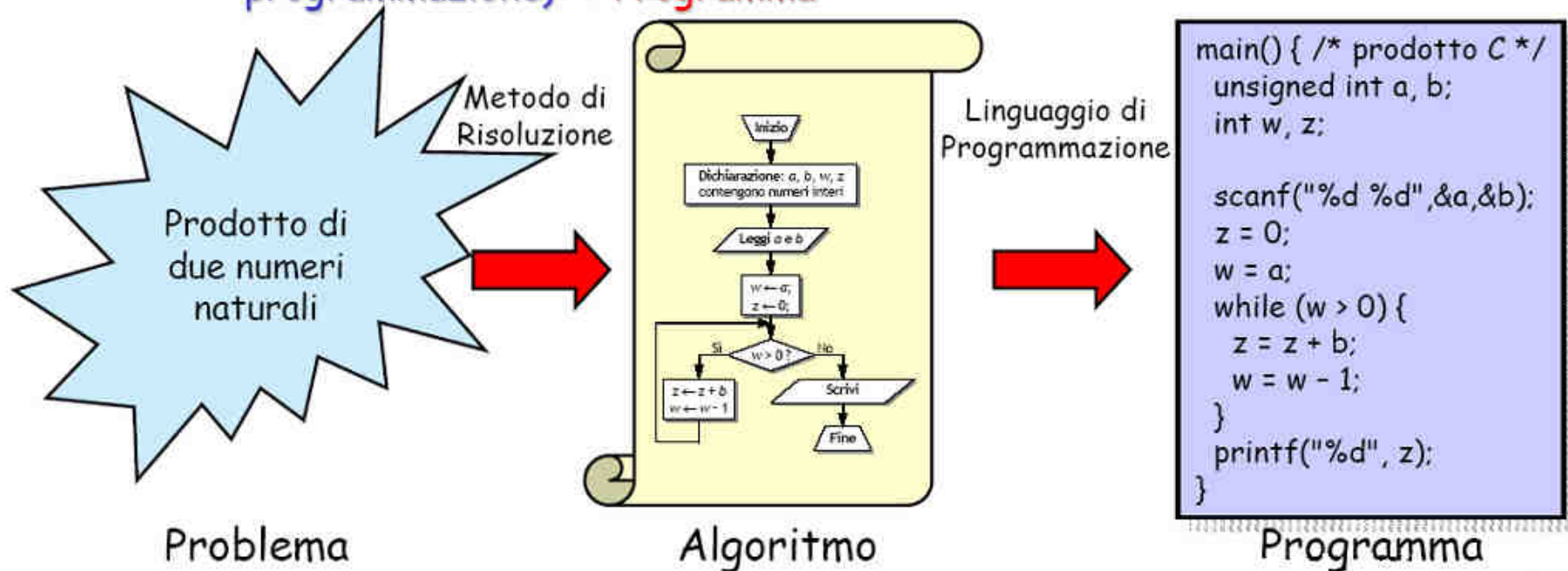
traduzione dell'algoritmo nel linguaggio di programmazione.

Esempio di Problema

- ❑ **Problema:** Risolvere l'equazione $ax+b=0$
- ❑ **Soluzione**
 1. leggi i valori di a e b
 2. calcola $-b$
 3. dividi quello che hai ottenuto per a e chiama x il risultato
 4. stampa x

Algoritmi e Programmi

- Passi per la risoluzione di un problema:
 - individuazione di un procedimento risolutivo
 - scomposizione del procedimento in un insieme ordinato di azioni
⇒ **Algoritmo**
 - rappresentazione dei dati e dell'algoritmo attraverso un formalismo comprensibile dal calcolatore (linguaggio di programmazione) ⇒ **Programma**



Dati e istruzioni

□ Tipi di dati

- Numeri naturali o interi o reali (1, -2, 0.34)
- Caratteri alfanumerici (A, B, ..)
- Dati logici o booleani (Vero, Falso)
- Array o vettore di n elementi ({1,2,3})

□ Istruzioni

- Operazioni di Input/Output (es. *leggi, scrivi*)
- Operazioni Aritmetico-logiche (es. $max = A + B$)
- Strutture di Controllo (es. *SE, RIPETI*)

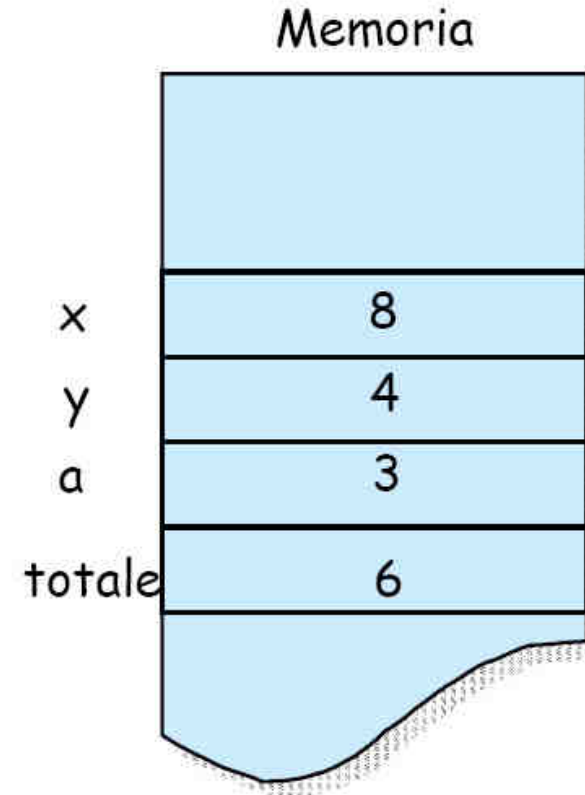
Algoritmi e Variabili

□ Gli algoritmi sono parametrici:

- producono un risultato che dipende da un insieme di dati di partenza;
- descrivono la soluzione non di un singolo problema, ma di una intera classe di problemi strutturalmente equivalenti.
- Esempi:
 - ✓ l'algoritmo per la moltiplicazione di due numeri specifica come effettuare il prodotto di *tutte* le possibili coppie di numeri;
 - ✓ l'algoritmo per la ricerca di un libro nello schedario della biblioteca vale per tutti i possibili libri;
 - ✓ ...

Algoritmi e Variabili

- ❑ Le istruzioni dell'algoritmo fanno riferimento a *variabili*
 - Somma x ad y
 - Se $a > 0$ allora
 - ...
 - Come in matematica una variabile è un sinonimo per indicare un dato
- ❑ **Variable**: Nome associato ad una locazione di memoria
 - Contenitore per dati
- ❑ **Astrazione delle celle di memoria**



Uso delle Variabili

- All'interno di espressioni,
 - l'esecutore usa il valore contenuto nelle variabili per calcolare il risultato dell'espressione,
 - per esempio $\text{var1} + \text{var2} \times \text{var3}$ oppure $\text{var1} / \text{var2} - \text{var3}$,
...
- in istruzioni di assegnamento/assegnazione
 - introdurre nel contenitore identificato dal nome della variabile il valore specificato a destra dell'assegnamento;
 - Esempi:
 - $r \leftarrow 35$ (assegna 35 alla variabile il cui nome è r),
 - $\text{pi} \leftarrow 3,14$
 - ...

Uso delle Variabili

- in istruzioni di assegnamento combinate con espressioni
 - assegna a una variabile il risultato ottenuto dalla valutazione di un'espressione
 - $\text{circ} \leftarrow 2 \times r \times \text{pi}$
 - ✓ il risultato dell'espressione $2 \times r \times \text{pi}$ viene calcolato utilizzando i valori contenuti nelle variabili r e pi e il risultato viene poi assegnato alla variabile circ ;
 - la stessa variabile può comparire in entrambi i lati dell'istruzione di assegnamento
 - ✓ $k \leftarrow k + 1$
 - ✓ il valore contenuto in k viene utilizzato per trovare il valore dell'espressione $k + 1$ che viene memorizzato come nuovo valore di k .

Rappresentazione degli algoritmi

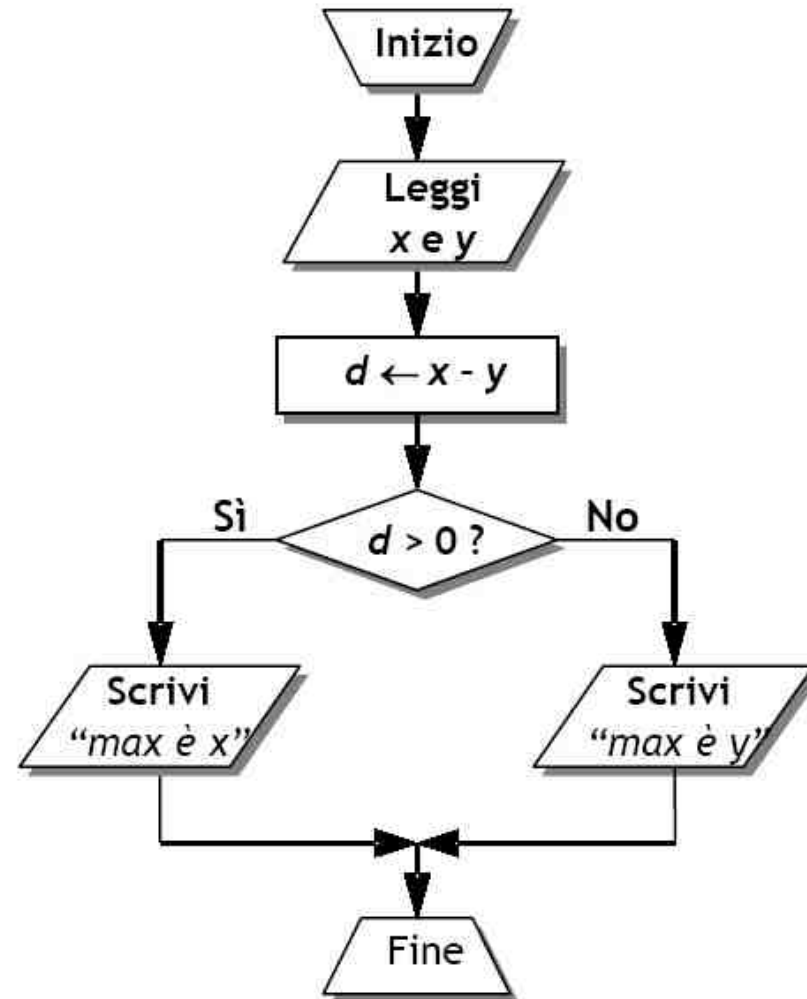
- ❑ Linguaggio naturale
- ❑ Diagramma a blocchi
- ❑ Pseudo codice
- ❑ Linguaggio di programmazione

Rappresentazione degli algoritmi

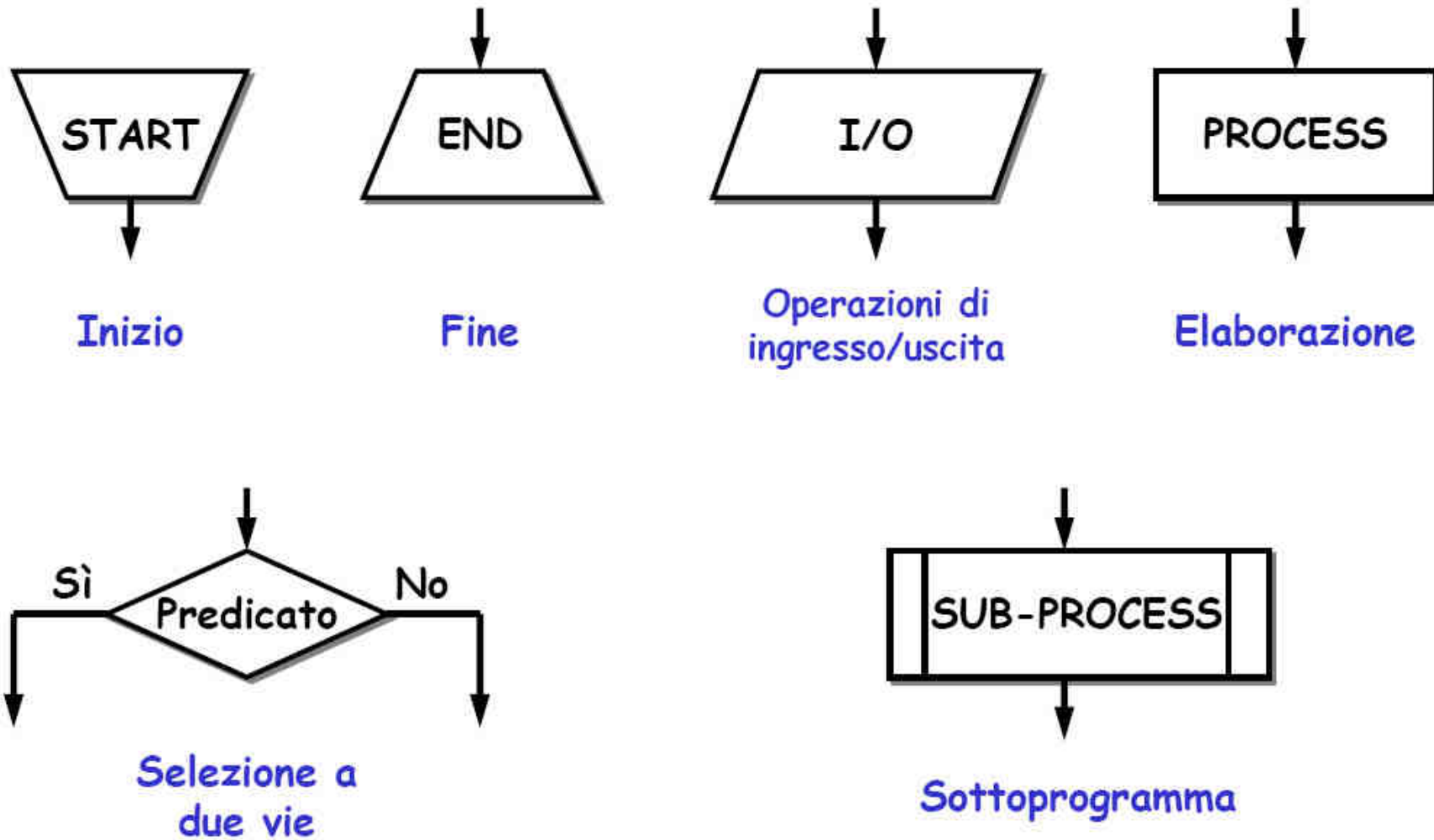
□ Linguaggio Naturale

- Sollevare il ricevitore
- Attendere il segnale di linea libera
- Comporre il numero
- ...

□ Diagramma di flusso

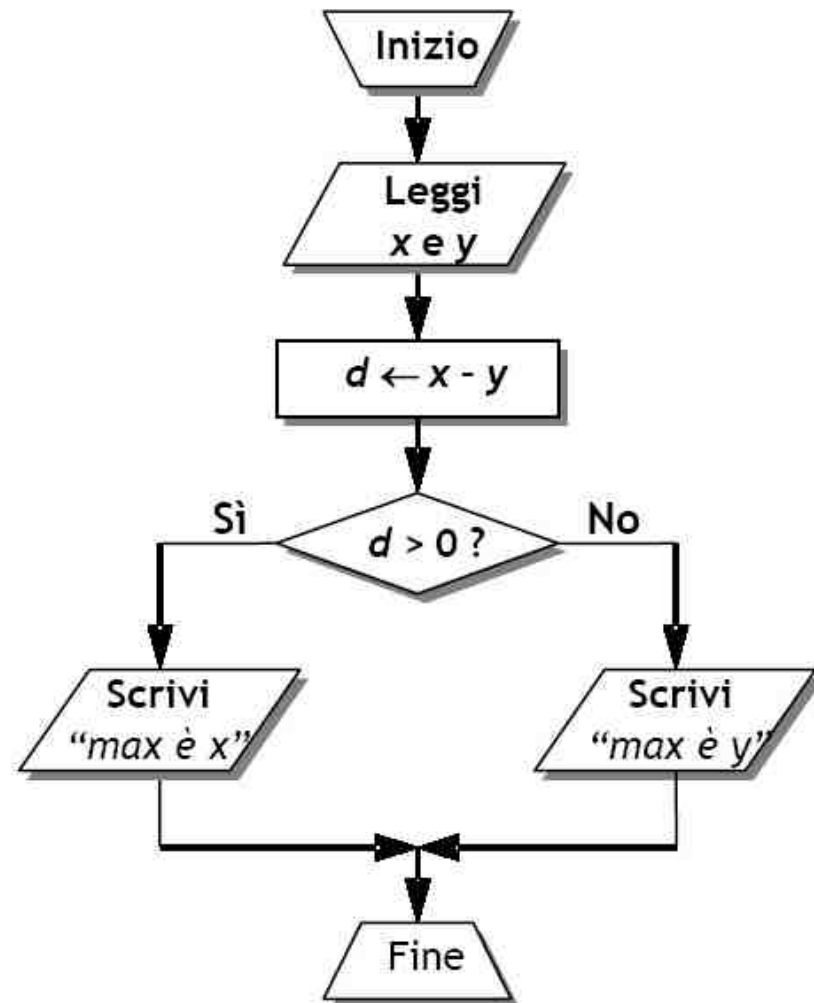


Diagrammi di flusso

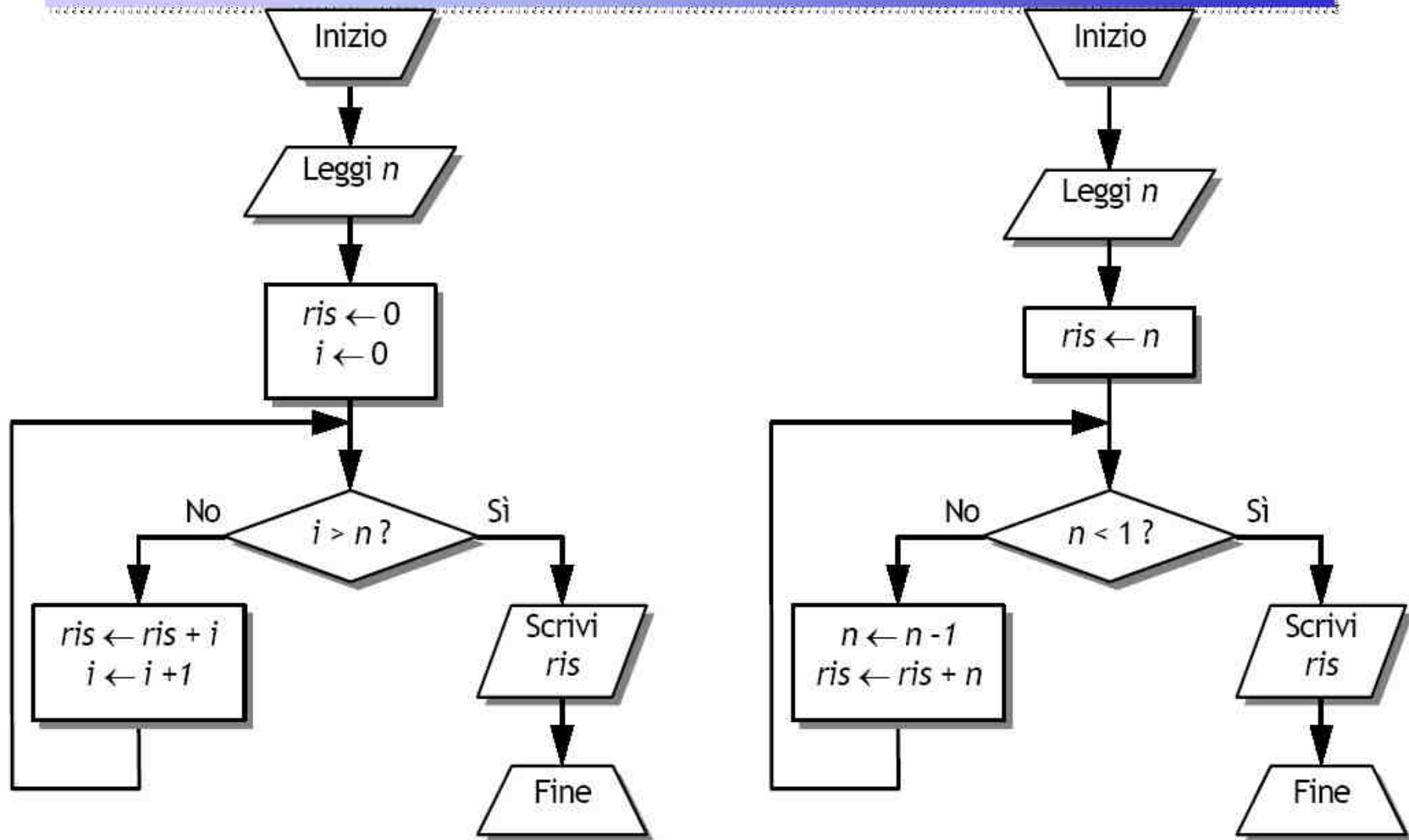


Esempio

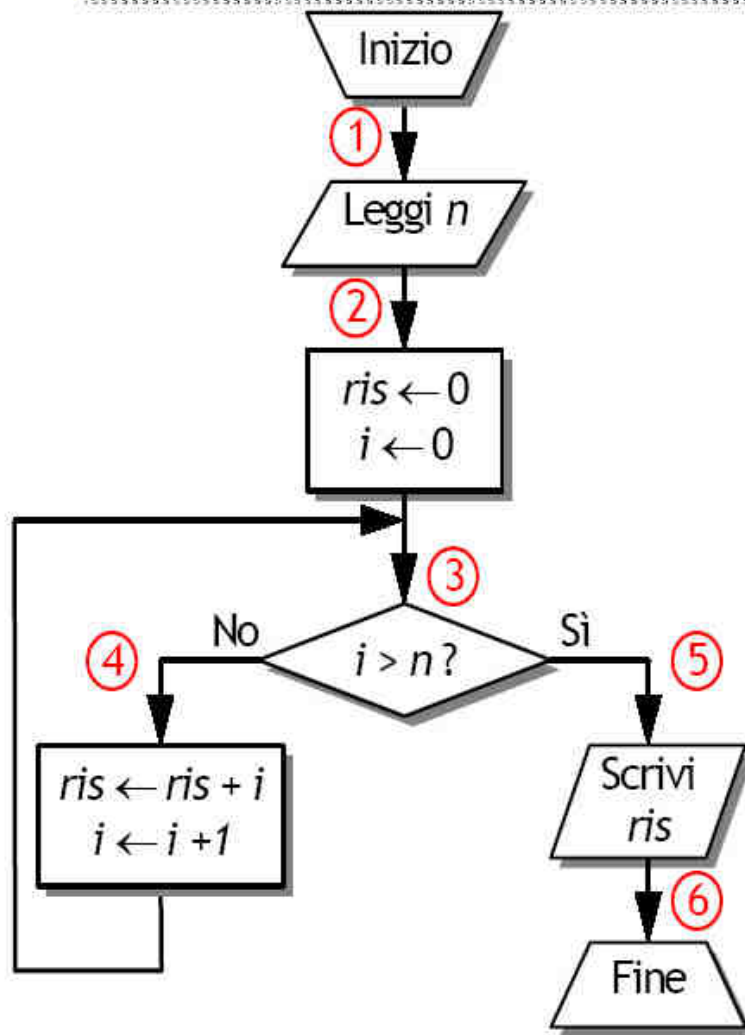
Esempio:
dati in ingresso due numeri
 x e y , si calcoli e stampi il
maggiore.



Somma dei primi N numeri naturali



Somma dei primi N numeri naturali

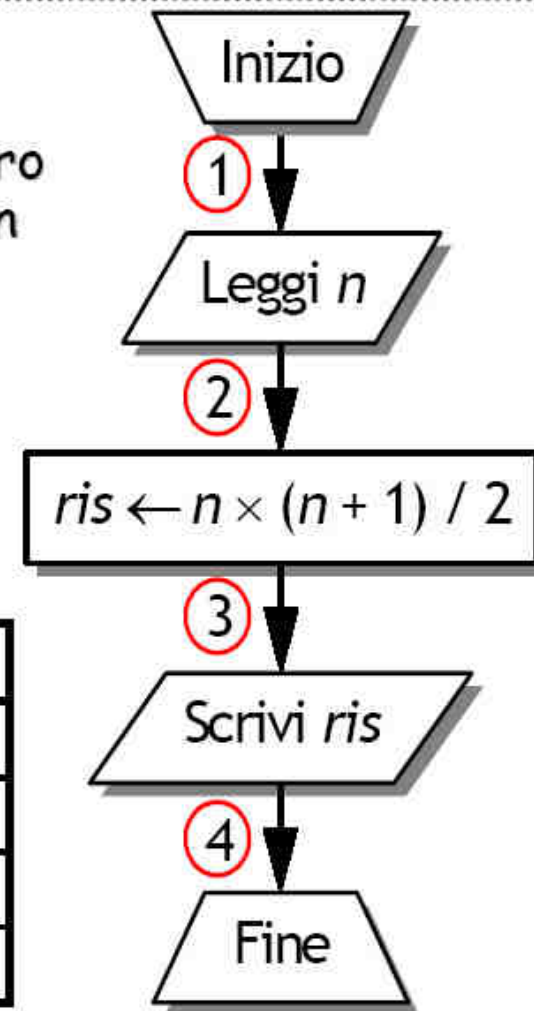


T	posiz.	n	i	ris	note
t ₀₁	①	??	??	??	Variabili non ancora definite
t ₀₂	②	4	??	??	Letto il valore 4 e inserito in n
t ₀₃	③	4	0	0	$i < n \Rightarrow$ posiz. 4
t ₀₄	④	4	0	0	
t ₀₅	③	4	1	0	$i < n \Rightarrow$ posiz. 4
t ₀₆	④	4	1	0	
t ₀₇	③	4	2	1	$i < n \Rightarrow$ posiz. 4
t ₀₈	④	4	2	1	
t ₀₉	③	4	3	3	$i < n \Rightarrow$ posiz. 4
t ₁₀	④	4	3	3	
t ₁₁	③	4	4	6	$i = n \Rightarrow$ posiz. 4
t ₁₂	④	4	4	6	
t ₁₃	③	4	5	10	$i > n \Rightarrow$ posiz. 5
t ₁₄	⑤	4	5	10	
t ₁₅	⑥	4	5	10	Stampato risultato (10)

Somma dei primi N numeri naturali

- ❑ Esiste infatti una soluzione analitica:
 $n \times (n + 1) / 2$
- ❑ L'algorithmo è molto più semplice (il numero di istruzioni da eseguire è costante e non dipende dal valore di n).

T	posiz.	N	ris	note
t ₀₁	①	??	??	Variabili non ancora definite
t ₀₂	②	4	??	Letto il valore 4 e inserito in n
t ₀₃	③	4	10	Calcolato il risultato $10=4 \times 5 / 2$
t ₀₄	④	4	10	Stampato risultato (10)



Prodotto di due numeri naturali

Dati

a, b interi positivi

w, z interi

Risoluzione

leggi a e b

$z \leftarrow 0$

$w \leftarrow a$

finché $w > 0$ ripeti

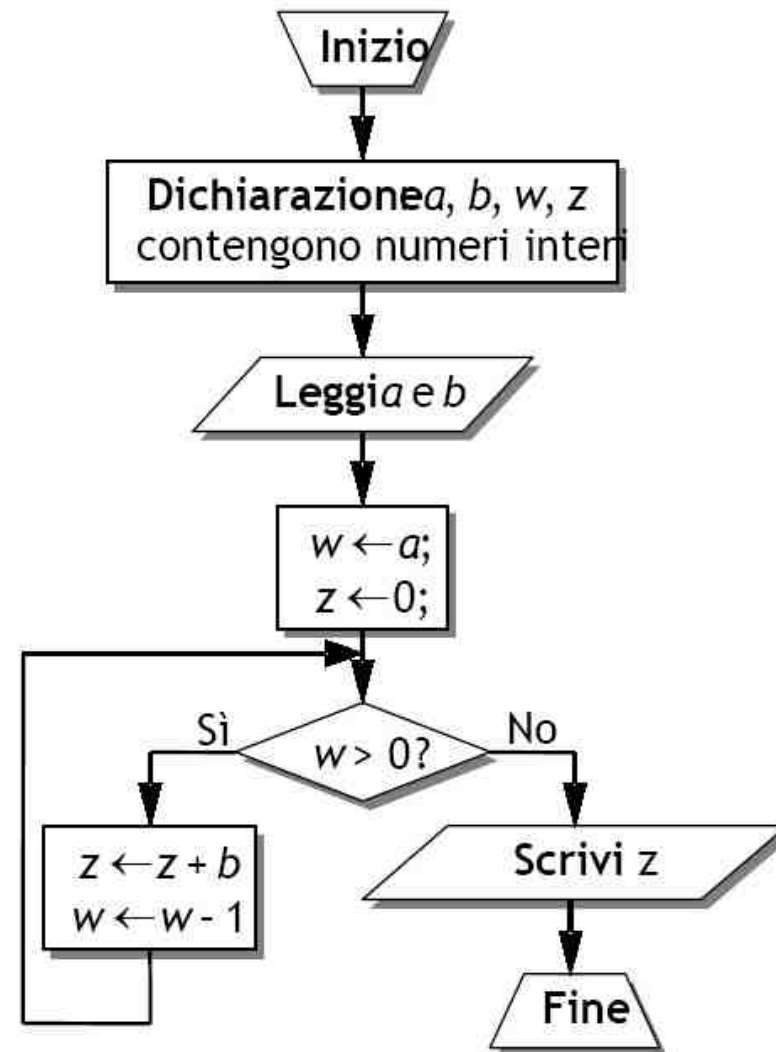
$z \leftarrow z + b$

$w \leftarrow w - 1$

fine ciclo

scrivi z

fine



Programmi e Programmazione

- ❑ **Programma:** È la descrizione di un algoritmo in un particolare linguaggio di programmazione.
- ❑ Quali "parole chiave" ?
- ❑ Quali dati ?
- ❑ Quali operazioni elementari ?
- ❑ Quali meccanismi di combinazione ?
- ❑ **Un linguaggio di programmazione** è una notazione formale per descrivere algoritmi che è comprensibile ad un calcolatore.

Linguaggi di Programmazione

- ❑ Ogni linguaggio è caratterizzato da:
 1. **sintassi**: l'insieme di regole formali per la scrittura di programmi in quel linguaggio, che dettano le modalità per costruire frasi corrette nel linguaggio stesso.
 2. **semantica**: l'insieme dei significati da attribuire alle frasi (sintatticamente corrette) costruite nel linguaggio.
- ❑ Una frase può essere sintatticamente corretta e tuttavia non aver alcun significato!
- ❑ Lo stesso può accadere per una istruzione.

Linguaggi di Programmazione

- ❑ Esistono diversi tipi di Linguaggi di Programmazione
- ❑ Linguaggi macchina e linguaggi assembler
 - ogni azione è indicata in codice binario o con operazioni molto semplici e "rudimentali" : **ADD X, Y** oppure **STORE A**
- ❑ Linguaggi imperativi (PASCAL, FORTRAN, C, BASIC, ...)
 - le azioni da compiere sono indicate in una sequenza che partendo dai dati si completa calcolando i risultati :
if a > 0 print ("valore positivo")
else print ("valore negativo");
- ❑ Linguaggi dichiarativi (logici - PROLOG, funzionali - LISP)
 - un programma è la definizione di una funzione o l'elenco delle regole logiche che portano a verificare una condizione.

Linguaggi di Programmazione

- Linguaggi orientati agli oggetti (C++, Java, Smalltalk,)
 - Sono basati sul concetto di **oggetto** software che rappresenta un oggetto del mondo reale (un numero, un archivio, un testo, una matrice).
 - I dati sono rappresentati come oggetti e le azioni da compiere come operazioni da effettuare sugli oggetti.
 - Di solito sono realizzati come estensione dei linguaggi imperativi.
 - Un programma modella un problema reale come una collezione di oggetti software che interagiscono.

Linguaggi di Programmazione e Astrazione

- Esistono linguaggi a vari livelli di astrazione

- Linguaggio Macchina

```
0100 0000 0000 1000
0100 0000 0000 1001
0000 0000 0000 1000
```

Sequenze di istruzioni macchina in binario direttamente eseguibili dal processore

- Linguaggio Assembler

```
LOAD X
ADD Y
STORE Z
```

Istruzioni hanno corrispondenza 1 a 1 con quelle macchina, ma vengono espresse con nomi simbolici

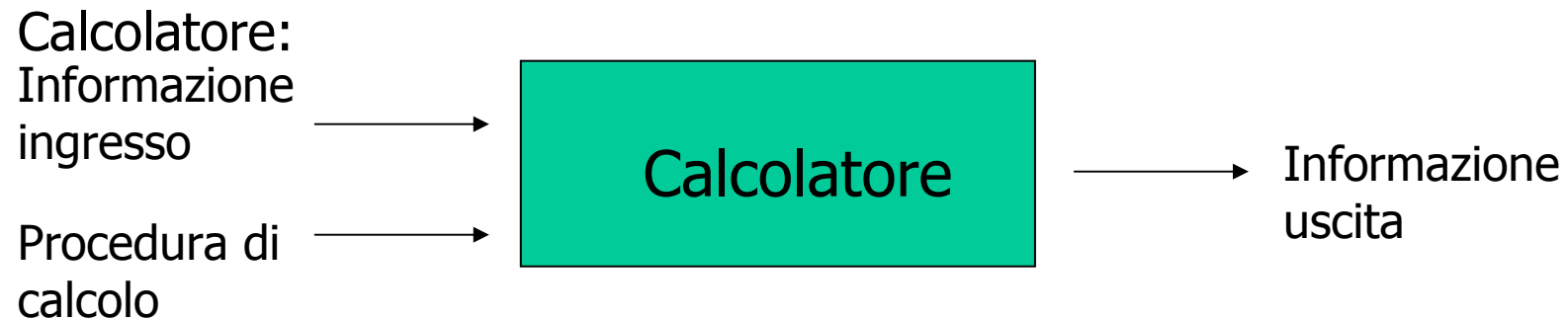
- Linguaggio di Alto Livello

```
main ( ) {
    int a=0;
    cout << a+5;
    return 0;
}
```

Indipendente dalla macchina.
Astrazione dati

Riassumendo

Elaborazione Automatica delle Informazioni

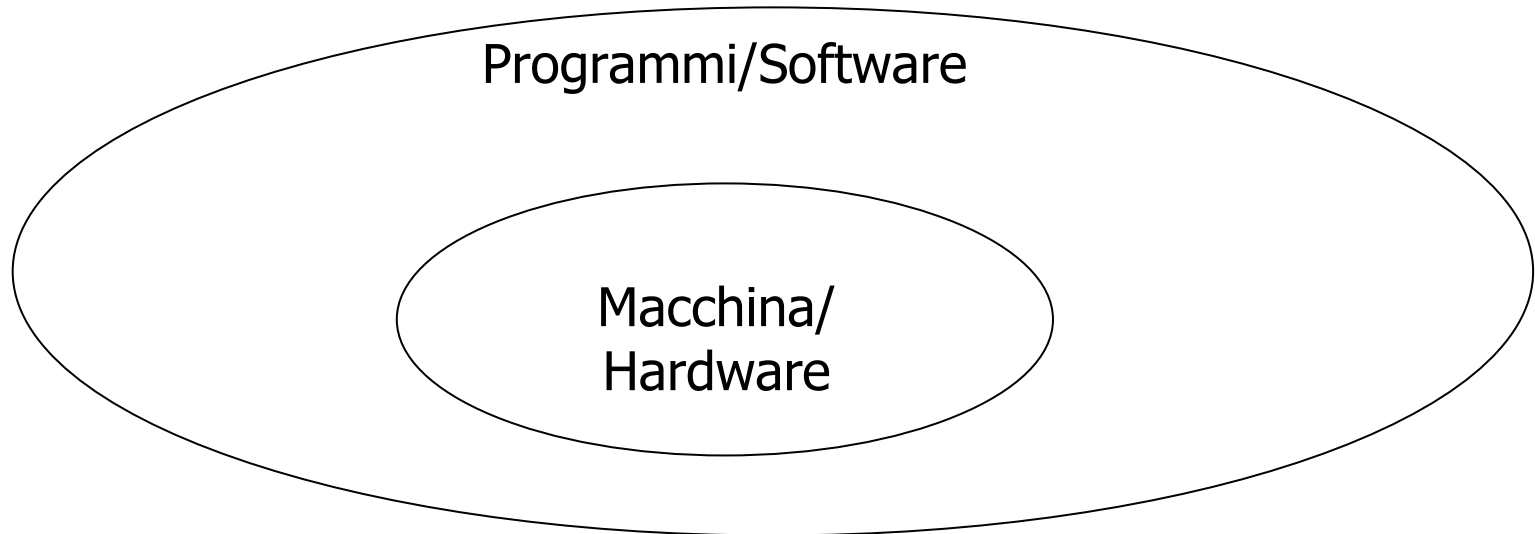


Algoritmo: *metodo risolutivo (sequenza ordinata di operazioni) per il calcolo del risultato a partire dai dati in ingresso.*

Linguaggio di programmazione: *linguaggio specializzato per rappresentare alla macchina algoritmi e dati.*

Programma: *specifica di un algoritmo in un linguaggio di programmazione.*

Hardware & Software



Hardware: struttura fisica del calcolatore

Software: programmi