# Informatica

## Esistono varie definizioni:

- "Scienza dei calcolatori elettronici (*Computer Science*)"
- "Scienza dell'informazione"

> "Scienza della rappresentazione, memorizzazione, ed elaborazione dell'informazione."

## Calcolatore Elettronico:

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

## Fanno parte dell'Informatica:

- metodi per la rappresentazione di procedimenti risolutivi
- studio dei linguaggi di programmazione
- studio dell'architettura dei calcolatori
- studio dei sistemi operativi
- metodi di calcolo numerico
- etc.

## Attenzione:

- concetto molto ampio di informazione (testi, dati numerici, immagini, suoni, eccetera);
- concetti molto ampi di rappresentazione (digitale, grafica, meccanica, sonora) ed elaborazione (trasformazione da una forma ad un'altra, estrazione di dati significativi, trasmissione da un punto ad un altro, eccetera).

# Perché l'Informatica

## Automatizzazione attività:

- Molte delle nostre attività intellettuali (quelle che, in sostanza elaborano l'informazione in modo meccanico) sono automatizzabili (es. somme, divisioni, etc. etc., preparazione delle matrici di stampa, preparazione di grafici).
- I calcolatori permettono di liberarci da queste attività e riescono a svolgerle in modo più veloce).

## Information Overloading:

- Viviamo in un ambiente sempre più ampio (dal villaggio al villaggio globale), abbiamo sempre più informazioni a disposizione (p.e., dal Gazzettino di Reggio alla CNN).
- Le nostre attività e le nostre decisioni devono basarsi su un substrato informativo sempre più ampio, che non possiamo riuscire a gestire (siamo esseri a "razionalità limitata")
- necessità di strumenti che ci aiutino nella elaborazione di informazioni, e nella estrazione di dati essenziali da tutta l'informazione che riceviamo.

## Information Access:

- Più informazione riusciamo ad accedere, più le nostre decisioni saranno razionali.

- Accesso veloce a informazioni remote (p.e. Internet)

# Il Calcolatore Elettronico

## Calcolatore elettronico:

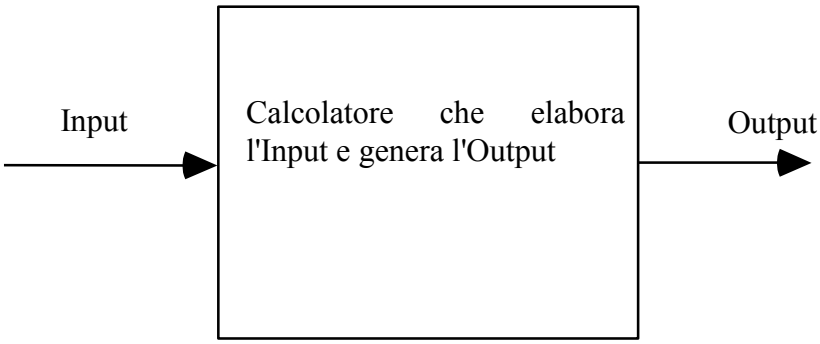
- e' uno strumento in grado di eseguire insiemi di *azioni* elementari.

- le azioni vengono eseguite su oggetti (*dati*) per produrre altri oggetti (*risultati*).

- L'esecuzione di azioni viene richiesta all'elaboratore attraverso opportune direttive, dette **istruzioni**.

## Programmazione

E` l'attività con cui si predispone l'elaboratore ad eseguire un particolare insieme di azioni su particolari dati (informazioni), allo scopo di risolvere un certo problema o di elaborare l'informazione secondo certi criteri.



# Cosa possiamo fare?

Le cose che siamo interessati a fare con un calcolatore sono di natura molto varia. Per esempio:

1) **Word Processing**. Memorizzare testi, gestire la grafica del testo, stampare il testo. Ruolo dell'elaboratore: memorizzare, elaborare il testo sulla base di comandi (p.e., cambia tipo di carattere) e sulla base di algoritmi (p.e., ritorno a capolinea quando finisce la riga, allineamento del testo).
2) **Basi di Dati**. Gestire grossi archivi informativi (p.e. anagrafe), memorizzare i dati, permettere il recupero veloce dei dati, permettere di estrarre informazioni globali sui dati (p.e., età media popolazione).
3) **Accesso Remoto**. trasmissione e recupero veloce di informazione (p.e., reti di calcolatori e Internet).
4) **Calcolo**. Risolvere problemi matematici complessi (ma anche semplici, perché no).
5) **Simulazioni.** Rappresentare in termini di informazioni una determinata situazione (virtualizzazione della realtà), elaborare questa sulla base di regole che simulano l'ambiente reale (anche per **intrattenimento**, p.e., videogames).

Tutto è in qualche modo riconducibile al concetto di **dati di input** (informazione sotto qualche forma e memorizzata da qualche parte) **elaborati** secondo una serie di azioni elementari dal calcolatore per produrre **dati di output** (diversa informazione sotto qualche forma e memorizzata da qualche parte).

# Elaborazione della Informazione come Risoluzione di un problema

Sapere che bisogna elaborare l'informazione in un certo modo non implica avere risolto il problema di sapere come questa elaborazione deve essere fatta dal calcolatore.

Dato un problema relativo alla elaborazione di informazione **BISOGNA**

- individuare un opportuno metodo di elaborazione che trasformi i dati iniziali nei corrispondenti risultati finali desiderati.

**Calcolatore=Esecutore di Azioni Elementari**

✍ Affinche` la risoluzione di un problema possa essere realizzata attraverso l'uso del calcolatore, tale processo deve poter essere definito come **sequenza di azioni elementari**.

# Problemi non Risolvibili

Non tutti i problemi di elaborazione sono risolvibili attraverso l'uso del calcolatore.

**Ad esempio:**

- il problema presenta infinite soluzioni o il metodo presenta infinite azioni (p.e. calcolo di pi greco)
- per alcuni dei problemi **non e` stato trovato** un metodo risolutivo (p.e., trovare tutte le coppie contigue di numeri primi)
- per alcuni problemi e' stato dimostrato che **non esiste** un metodo risolutivo automatizzabile (p.e., trisecare un angolo con solo riga e compasso)

- Noi ci concentreremo sui problemi che, ragionevolmente, ammettono un metodo risolutivo (**funzioni calcolabili**).

Uno degli obiettivi del Corso, tra gli altri, è presentare le **tecnologie** e le **metodologie** di programmazione:

**Tecnologie:** strumenti per la realizzazione di programmi.

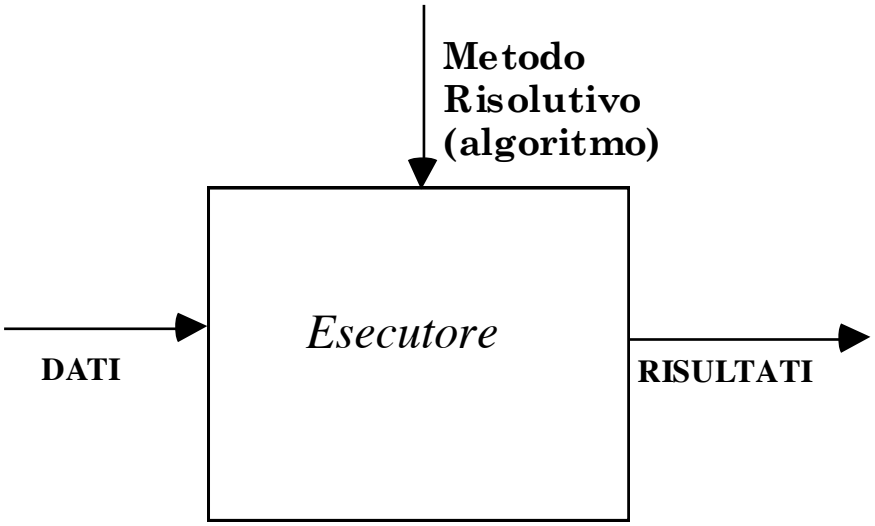
**Metodologie:** metodi per l'utilizzo in modo corretto ed efficace delle tecnologie di programmazione.

# Algoritmo

**L'algoritmo** e` l'insieme ordinato delle azioni che risolve un dato problema.

- l'algoritmo descrive il metodo risolutivo attraverso un insieme ordinato di azioni.

- l'esecuzione delle azioni secondo l'ordine specificato dall'algoritmo, a partire dai dati di ingresso, consente di ottenere i risultati relativi alla soluzione del problema.




☞ Si fa riferimento ad un "**esecutore**": con questo termine si intende una macchina astratta (non necessariamente un calcolatore) in grado di eseguire le azioni specificate dal metodo risolutivo.

# Proprietà fondamentali dell'Algoritmo

1• **Eseguibilità**: ogni "istruzione" deve essere eseguibile da parte dell'*esecutore* dell'algoritmo;
2• **Non Ambiguità**: ogni azione deve essere univocamente interpretabile dall'*esecutore*
3• **Finitezza**: il numero totale di azioni da eseguire, per ogni insieme di dati di ingresso, e' finito.

**Quindi, l'algoritmo deve:**

- essere applicabile a qualsiasi insieme dei dati di ingresso appartenenti al **dominio di definizione dell'algoritmo**;

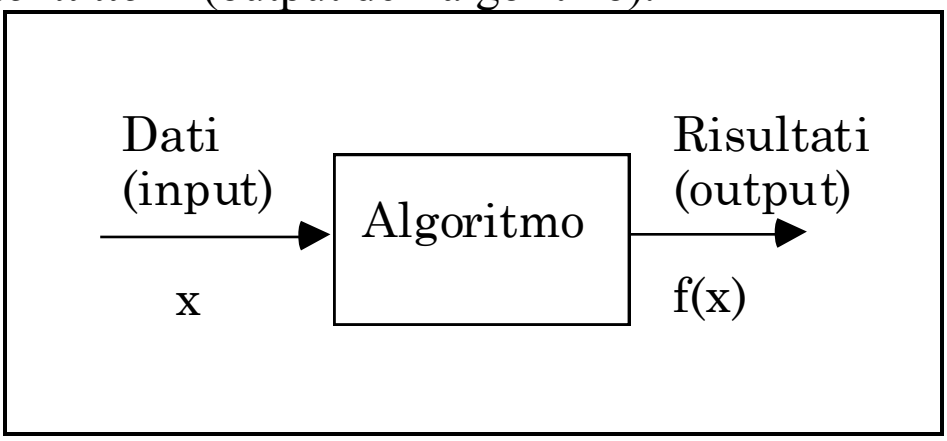
- essere costituito da operazioni appartenenti ad un determinato **insieme di operazioni fondamentali** (sistema formale);

- le regole che costituiscono l'algoritmo non devono essere ambigue, cioe` devono essere interpretabili in modo **univoco** qualunque sia la persona o la "macchina" che le legge

# Algoritmo

- Altre proprieta` *desiderabili*:
  - generalita`
  - efficienza
  - determinismo
  - ...

- Un algoritmo puo` essere visto come il procedimento di calcolo di una funzione che mappa uno o piu` valori di un ***dominio A*** (input dell'algoritmo) in un valore del ***codominio B*** (output dell'algoritmo).



# Algoritmo: un po' di storia

- Il termine "algoritmo" deriva dal nome del matematico arabo *Al-Khowarzimi* del IX secolo d.c. che per primo suggeri` un metodo per sommare due numeri rappresentati nel sistema numerico Hindu e contribui` alla fondazione dell'algebra.

- Nel medioevo il termine *algorismus* servi` ad indicare il complesso di operazioni nel calcolo numerico con numeri arabi.

- Attualmente, con il temine **algoritmo** si indica la sequenza finita di passi effettuabili da una macchina per risolvere una classe di problemi in tempo finito.

# Algoritmi e Programmi

- Ogni elaboratore e' una macchina in grado di eseguire azioni elementari su oggetti detti **dati**.

- L'esecuzione delle azioni e' richiesta all'elaboratore tramite comandi elementari chiamati **istruzioni**.

- Le istruzioni sono espresse attraverso un opportuno formalismo:
  il **linguaggio di programmazione**.

Es:

```
SUM A,B
read(x)
scanf("%d",&Y)
```

- La formulazione testuale di un algoritmo in un linguaggio comprensibile ad un elaboratore e` detta **programma**.

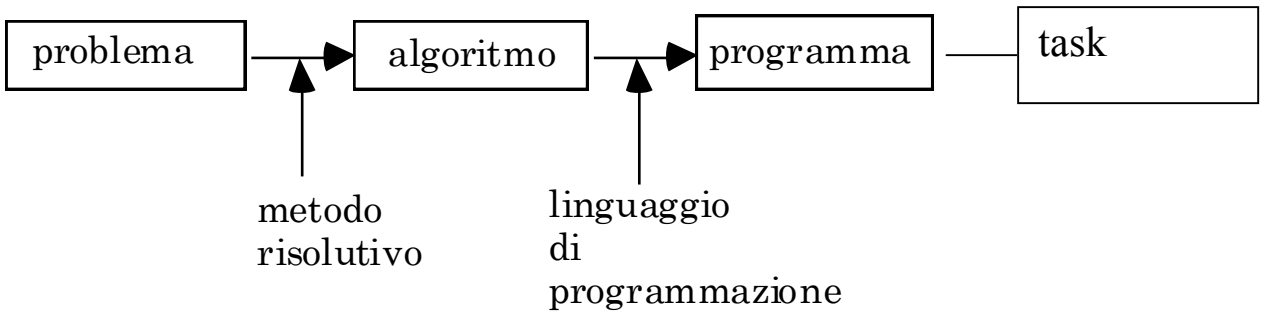
```
main()
{int A, B;

printf(“Immettere i dati (A,B): “);
scanf(“%d %d”, &A, &B);
printf(“Risultato: A+B=%d\n”, A+B);
}
```

# Algoritmi e Programmi

Dato un problema, la sua soluzione puo` essere ottenuta mediante l'uso del calcolatore, compiendo i seguenti passi:

1• individuazione di un procedimento risolutivo
2• scomposizione del procedimento in insieme ordinato di azioni **--> ALGORITMO**
3• rappresentazione dei dati e dell'algoritmo attraverso un formalismo comprensibile per l'elaboratore (linguaggio di programmazione **--> PROGRAMMA**
**4-** esecuzione di un programma sul calcolatore → processo o "task"




Il procedimento e' complesso e non e' univoco.

# Algoritmi: Alcuni Esempi

## Il Problema della capra, del lupo e del cavolo:

- Porta la capra sull'altra sponda;
- Torna indietro
- Porta il cavolo sull'altra sponda
- Porta indietro la capra
- Porta il lupo sull'altra sponda
- Torna indietro
- Porta la capra sull'altra sponda

## Soluzione dell'equazione ax +b=0

- leggi i valori di a e b
- calcola -b
- dividi quello che hai ottenuto per a e chiama x il risultato
- stampa x

**Stabilire se una parola P viene alfabeticamente prima di una parola Q**

- **leggi** P,Q
- **ripeti quanto segue:**
  **se** prima lettera di P < prima lettera Q
  **allora** scrivi vero
  **altrimenti se** prima lettera P > Q
  **allora** scrivi falso
  **altrimenti** (le lettere sono =)
  togli da P e Q la prima lettera

  **fino** a quando hai trovato le prime lettere diverse.

**Calcolo del massimo di un insieme:**

- Scegli un elemento come massimo provvisorio *max*
- Per ogni elemento *i* dell'insieme: se *i>max* eleggi *i* come nuovo massimo provvisorio max
- Il risultato e' *max*

**NOTA**: si utilizzano **variabili** cioe' nomi simbolici usati nell'algoritmo per denotare dati.

# 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

Due algoritmi equivalenti:

- forniscono lo stesso risultato
- possono avere differente efficienza
- possono essere profondamente diversi

# Algoritmi Equivalenti: Esempio

## Calcolo del massimo comun divisore fra m ed n

- Algoritmo **a:**

  - Calcola l'insieme I dei divisori di m
  - Calcola l'insieme J dei divisori di n
  - Calcola l'insieme dei divisori comuni K (I∩J)
  - Calcola il massimo in K: questo e' il risultato

- Algoritmo **b**: si basa sul **metodo di Euclide**:

  **mcd(m,n) = m** (oppure n) se m=n
  **mcd(m,n) = mcd(m-n, n)** se m>n
  **mcd(m,n) = mcd(m, n-m)** se m<n

  **Algoritmo**:
  - Finche' m e` diverso da n esegui le seguenti azioni:
    se m>n sostituisci a m il valore (m-n)
    altrimenti sostituisci a n il valore (n-m)
  - Il massimo comun divisore e' n

  **Gli algoritmi a e b sono equivalenti.**

# Rappresentazione di Algoritmi: Diagrammi di flusso

E' un formalismo che consente di rappresentare graficamente gli algoritmi.

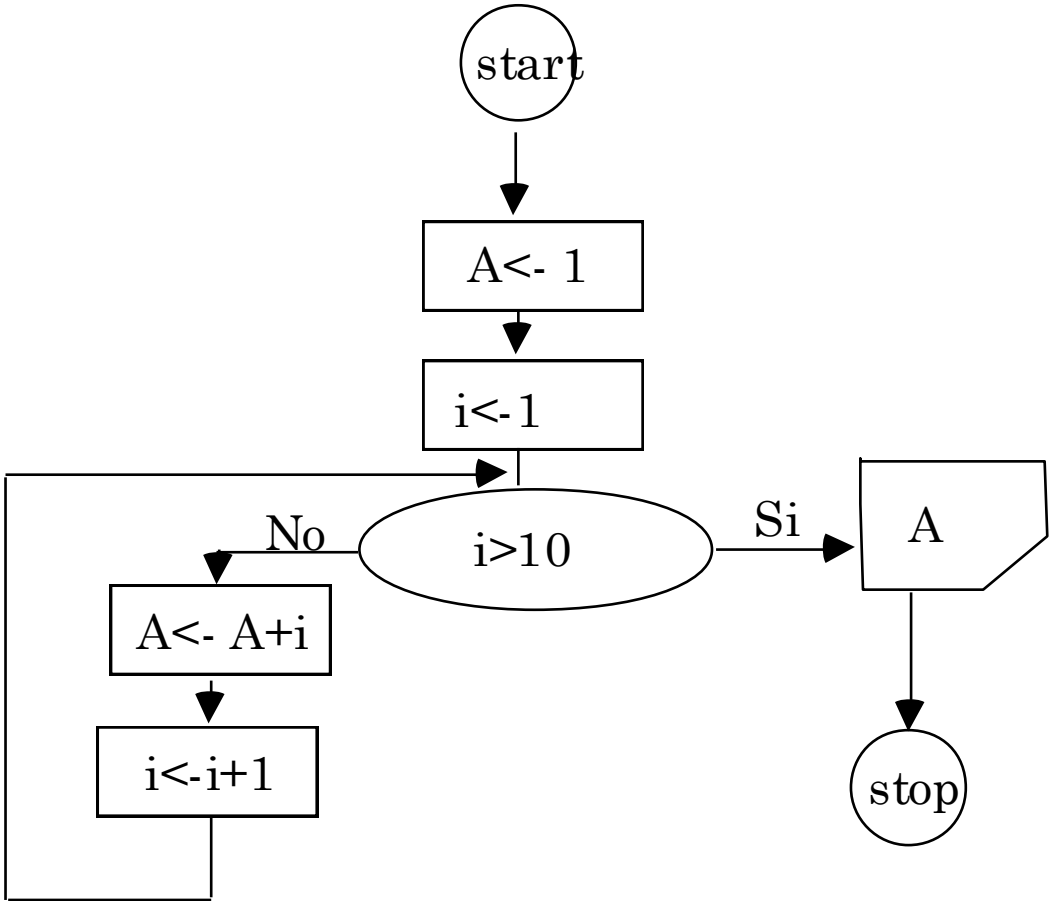
- un **diagramma** di flusso descrive le azioni da eseguire ed il loro ordine di esecuzione.

- ogni azione elementare corrisponde ad un simbolo grafico (blocco) diverso (sono convenzioni non universali).

- ogni blocco ha un ramo in ingresso ed uno o piu` rami in uscita; collegando tra loro i vari blocchi attraverso i rami, si ottiene un diagramma di flusso

- un diagramma di flusso appare, quindi, come un insieme di blocchi di forme diverse che contengono le istruzioni da eseguire, collegati fra loro da linee orientate che specificano la sequenza in cui i blocchi devono essere eseguiti (flusso del controllo di esecuzione).

# Diagrammi di Flusso




- L'insieme dei dati di ingresso e dei risultati vengono rappresentati attraverso dei nomi simbolici, detti **variabili (ad esempio, A)**

- Puo` essere inoltre necessario introdurre delle variabili "temporanee" (ad esempio, i), necessarie alla risoluzione del problema: tali variabili vengono anch'esse rappresentate da nomi simbolici.

I diagrammi di flusso hanno notevoli limiti per la soluzione di problemi di una certa complessita`.

# Diagrammi di Flusso

## Valori:

**Numerici**: interi e reali
**Alfanumerici**(stringhe): "AAAA","C.Colombo"
**Logici**: Vero e Falso

## Grandezze:

- **Costanti**: Quantita` note a priori, il cui valore non cambia durante l'esecuzione

- **Variabili**: grandezze rappresentate da un nome simbolico il cui valore puo` cambiare durante l'esecuzione dell'algoritmo.

# Diagrammi di Flusso

## Espressioni:

- Sequenze di variabili e costanti combinate fra loro mediante operatori (ad es, operatori **aritmetici:** +, -, *, /); ad esempio:

  s + r *5
  A
  100

- Nella valutazione di una espressione, si sostituisce ad ogni variabile il suo valore attuale e si eseguono le operazioni secondo un ordine prestabilito da regole di precedenza (possono comparire parentesi).

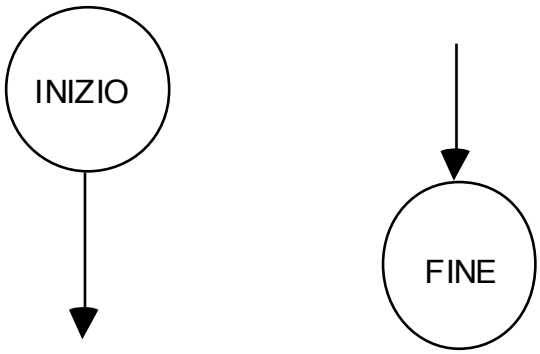
- A tutte le variabili che compaiono nell'espressione deve essere stato associato un valore prima della valutazione dell'espressione

- Espressioni relazionali e logiche: danno come risultato vero o falso
  (> minore , < maggiore, = = uguale, != diverso)

# Istruzioni (blocchi) fondamentali

## Inizio e fine esecuzione (start e stop)

**Inizio** e` il blocco da cui deve iniziare l'esecuzione (uno solo). Il blocco **fine** fa terminare l'esecuzione dell'algoritmo (almeno uno).

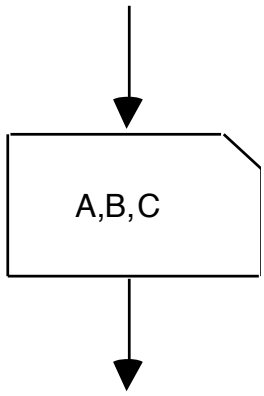



☞ A questi blocchi non corrisponde alcuna azione.

# Ingresso (lettura, read, input)

**Esecuzione dell'istruzione:**
Si ricevono dall'unita` di ingresso (per esempio, la tastiera) tanti valori quante sono le variabili specificate all'interno del blocco, e si assegnano nello stesso ordine alle variabili.

- A, B, C sono nomi di **variabili**.

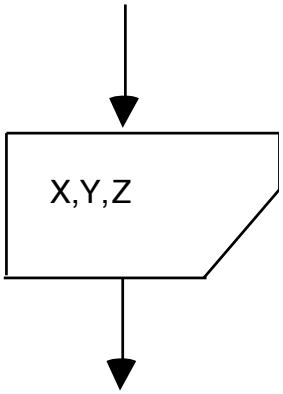



☞ *"Leggi i tre valori dati in ingresso, ed assegnali rispettivamente alle variabili A, B, e C."*

# Uscita (stampa, type, print, output)

## Esecuzione:

Si calcolano i valori delle espressioni e si trasmettono all'unita' di uscita (ad esempio, il video).




- X, Y, Z sono espressioni

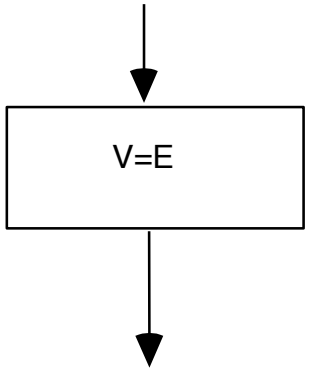
☞ *"Calcola i valori delle espressioni X, Y e Z, e trasmettili in uscita."*

**N.B.** I valori di X, Y, Z non vengono alterati dall'esecuzione del blocco

# Assegnamento

## Esecuzione:

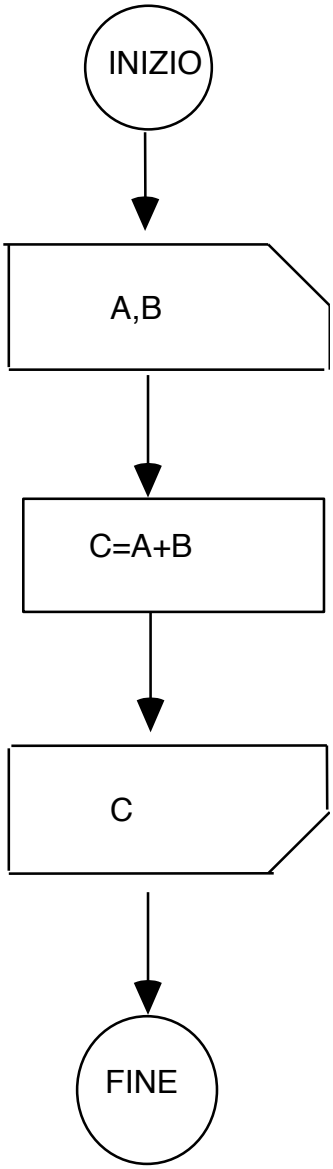
Si calcola il valore dell'espressione a destra del simbolo "=" e lo si assegna alla variabile indicata a sinistra del simbolo "=" (con eventuale perdita del valore precedente di V)




- V e` il nome di una variabile, E e` una espressione.

☞ *"Calcola il valore dell'espressione E e assegnalo alla variabile V"*

# Esempio (sequenza):

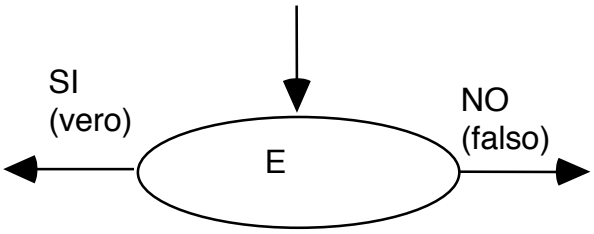


# Condizioni o salti condizionati

## Esecuzione:

Si valuta la condizione specificata all'interno del blocco: se e` verificata, si prosegue con la linea di flusso contrassegnata da "SI", altrimenti (se non e` verificata) si prosegue per il ramo etichettato con "NO".

- E e` un'espressione relazionale o logica (ritorna valore **vero**, oppure **falso**).

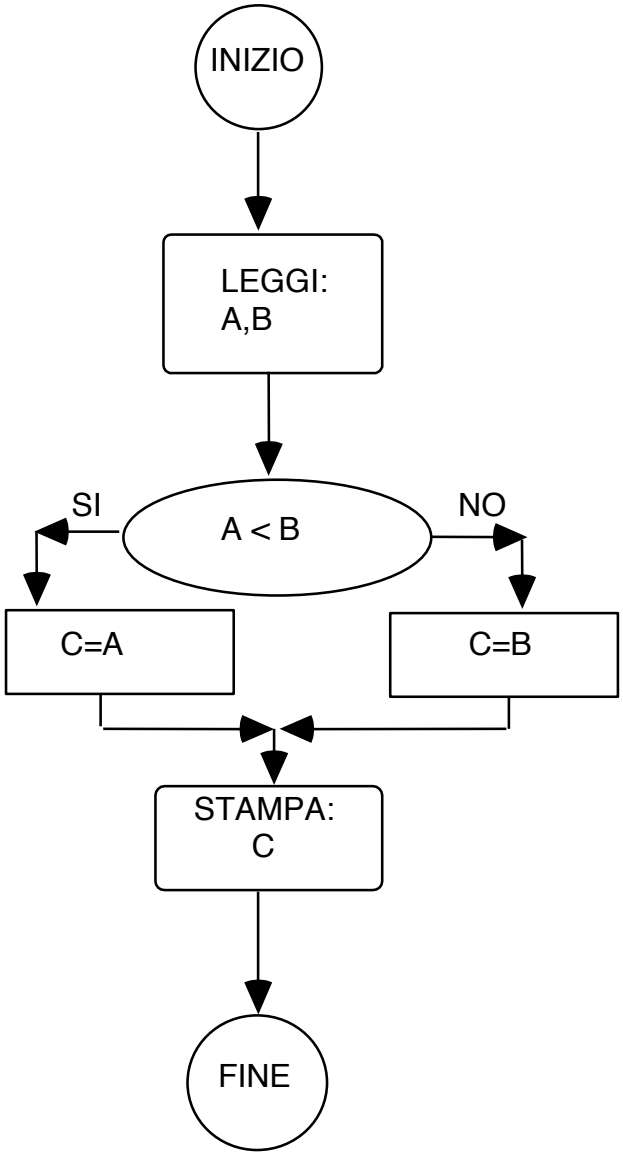



☞ *"Calcola il valore dell'espressione E: se e` vero, prosegui per il ramo SI, altrimenti prosegui per il ramo NO "*

Il blocco condizione e` l'elemento di base per realizzare **alternative** e **ripetizioni**.

# Alternativa

Esprime la scelta tra due possibili azioni (o sequenze di azioni) mutuamente esclusive:



# Iterazione o Ripetizione

Esprime la ripetizione di una sequenza di istruzioni.

Nel caso piu` generale (ripetizione enumerativa), e` costituita da:

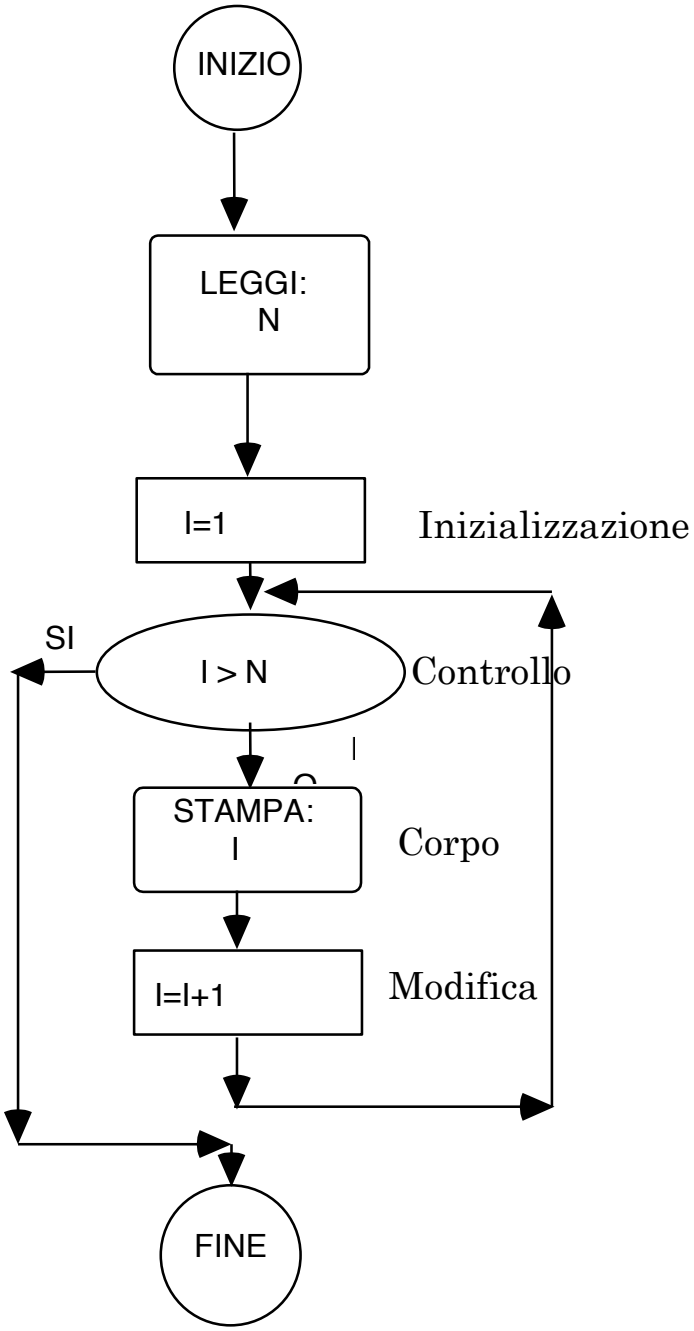
- **Inizializzazione**: assegnazione dei valori iniziali alle variabili caratteristiche del ciclo (viene eseguita una sola volta);

- **Corpo**: esecuzione delle istruzioni fondamentali del ciclo che devono essere eseguite in modo ripetitivo;

- **Modifica**: modifica dei valori delle variabili che controllano l'esecuzione del ciclo (eseguito ad ogni iterazione);

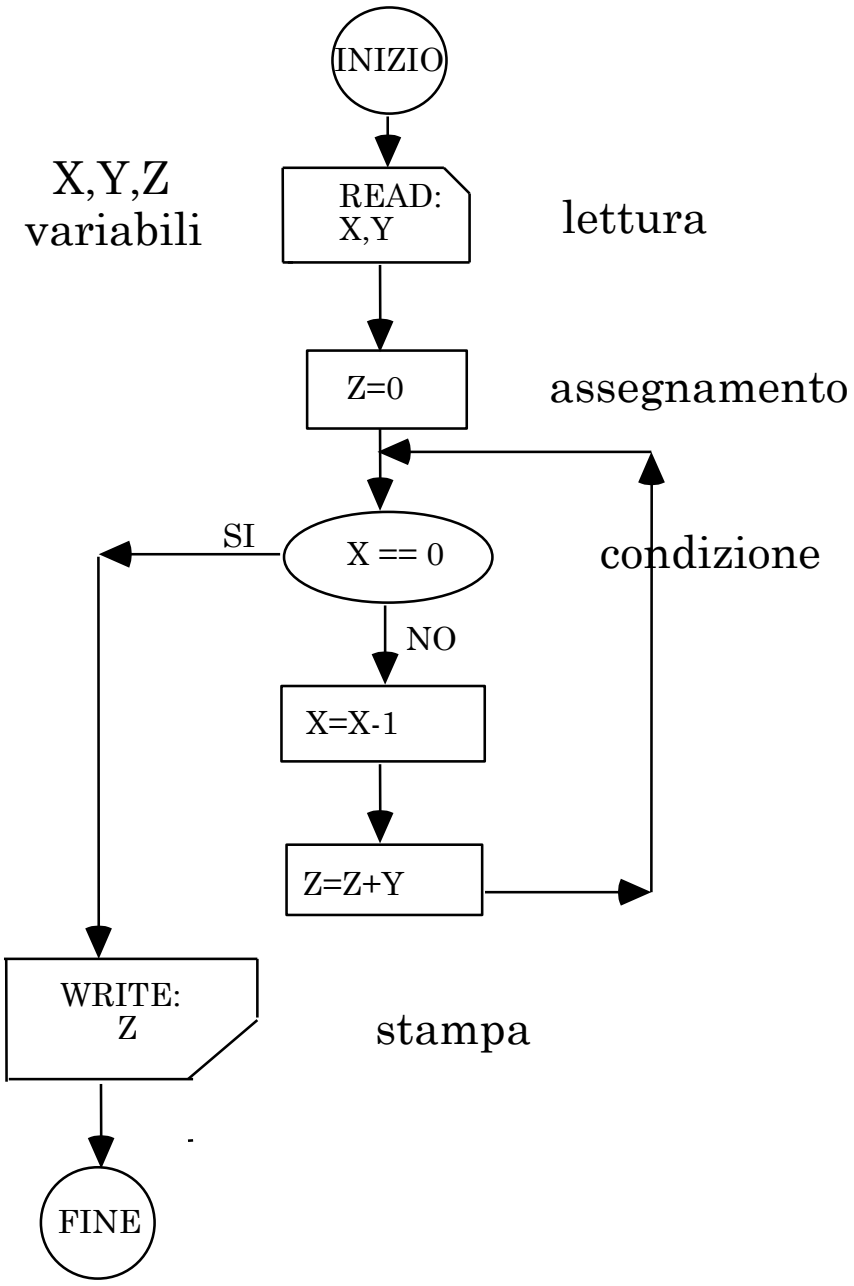
- **Controllo**: determina, in base al valore delle variabili che controllano l'esecuzione del ciclo se il ciclo deve essere ripetuto o meno.

# Iterazione


INIZIO
LEGGI:
N
I=1
Inizializzazione
SI
I > N
Controllo
I
STAMPA:
I
Corpo
I=I+1
Modifica
FINE

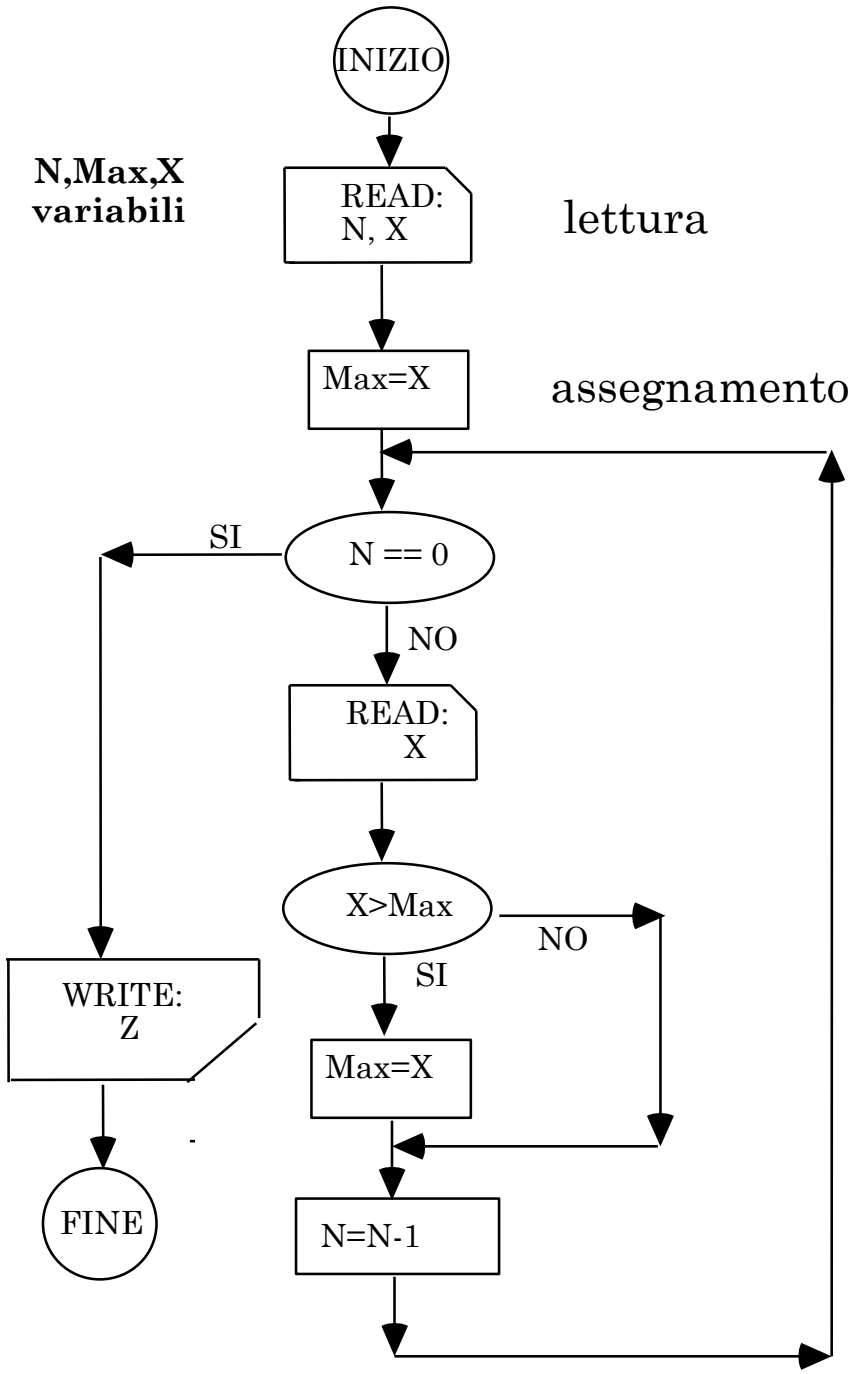
## Esempio:

Algoritmo che calcola il prodotto come sequenza di somme (si suppone Y>0, X≥0).



## Esempio:

Algoritmo che calcola il massimo di una sequenza di numeri interi.



# STRUTTURE ALGORITMICHE

In generale, qualsiasi algoritmo si compone attraverso le seguenti strutture:

- **concatenazione**, istruzioni di lettura, scrittura o assegnamento eseguite in sequenza
- **alternativa**
- **iterazione**

Chiaramente, la complessità di un algoritmo potrà essere alta o bassa, ma sempre in ogni caso sarà esprimibile attraverso queste strutture base (composte in modo varie, come componiamo pezzi di lego), che formano la cosiddetta base della programmazione "strutturata".