

# ALGORITMI

In informatica e matematica, il termine algoritmo indica un procedimento che risolve un determinato problema attraverso un numero finito di passi (cioè un numero limitato di operazioni).

Un problema risolvibile mediante un algoritmo si dice *computabile*.

Il termine "*algoritmo*" deriva dalla trascrizione latina del nome del matematico persiano Al - Khwarizmi, che è considerato uno dei primi autori ad aver fatto riferimento a questo concetto.

Deve essere:

**Effettivo:** cioè deve avere un punto di partenza e uno di fine, ed ogni operazione deve dare lo stesso risultato ogni volta che si hanno le stesse condizioni

**Definito:** cioè che devono essere presi in considerazione tutti i casi che un problema può presentare

**Non ambiguo:** Ogni passaggio deve essere interpretato da chiunque nello stesso modo

**Generale:** cioè che deve poter essere utilizzato per tutti gli stessi tipi di problemi, quest'ultimo insieme si chiama dominio dell'algoritmo

**Finito:** cioè si deve poter raggiungere il risultato con un numero limitato di operazioni

Il **costo computazionale** di una funzione/algoritmo/programma è un costo definito in termini di risorse di calcolo; le risorse di calcolo fondamentali sono due:

*quantità di tempo necessario alla computazione (tempo)*

*quantità di memoria utilizzata (spazio)*

A che serve stabilire il costo computazionale di una funzione/programma/algoritmo? Serve a *confrontare diverse funzioni/algoritmi/programmi che risolvono lo stesso problema, in modo da scegliere il più efficiente.*

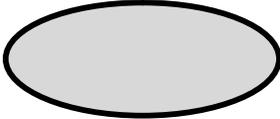
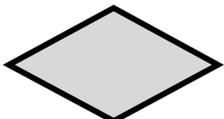
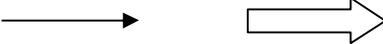
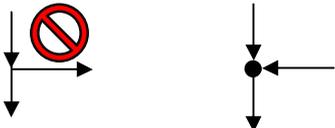
## DIAGRAMMI A BLOCCHI

I diagrammi a blocchi o diagrammi di flusso (flow charts) sono diagrammi a blocchi; un diagramma o schema si definisce a **blocchi quando ogni parte che lo compone** svolge una sua propria funzione in maniera indipendente dagli altri blocchi. Sono strumenti molto utilizzati in informatica per una chiara e semplice esposizione in forma grafica degli algoritmi, sono uno strumento fondamentale per il programmatore (servono per scomporre in maniera sistematica

il problema), danno una visione immediata d'insieme e facilitano la correttezza logica del problema; hanno anche una funzione comunicativa, infatti è molto più semplice capire un algoritmo attraverso uno schema a blocchi che leggerne una sua traduzione in un linguaggio di programmazione specifico; un diagramma di flusso deve avere i seguenti requisiti:

deve esistere un solo blocco di inizio, deve esistere un solo blocco di fine, può avere solo un numero finito di blocchi di controllo, può avere solo un numero finito di blocchi di ingresso/uscita e blocchi operativi, dall'inizio devo poter arrivare alla fine qualunque sia il percorso che seguo

## Simboli e regole di costruzione

	<p>blocco di inizio/fine (può avere solo una linea in uscita nel caso di blocco di inizio, può avere solo una linea in ingresso nel caso di blocco di fine, da ogni blocco iniziale si deve poter raggiungere attraverso un insieme non vuoto di blocchi il blocco di fine)</p>
	<p>blocco operativo aritmetico/logico (può avere solo una linea in ingresso e una sola linea in uscita)</p>
	<p>Chiamata a subroutine (sottoprogramma) usato per richiamare, senza riscriverlo tutte le volte, una parte del diagramma di flusso. Usato quando un insieme di blocchi si ripete più volte all'interno del diagramma di flusso, o quando un insieme di blocchi è usato da più diagrammi di flusso nello stesso modo</p>
	<p>blocco di ingresso/uscita (può avere solo una linea in ingresso e una sola linea in uscita)</p>
	<p>Blocco di selezione/controllo/condizione (deve avere solo una linea di ingresso e due uscite, chiamate: SI/NO oppure VERO/FALSO oppure CONDIZIONE A/CONDIZIONE B)</p>
	<p>Linee di flusso (una linea di flusso non si può diramare, più linee di flusso si possono collegare a uno stesso punto)</p>
	<p>Connessione a un punto qualsiasi del diagramma (dove confluiscono due o più flussi)</p>

L'algoritmo per il calcolo delle radici di secondo grado è:

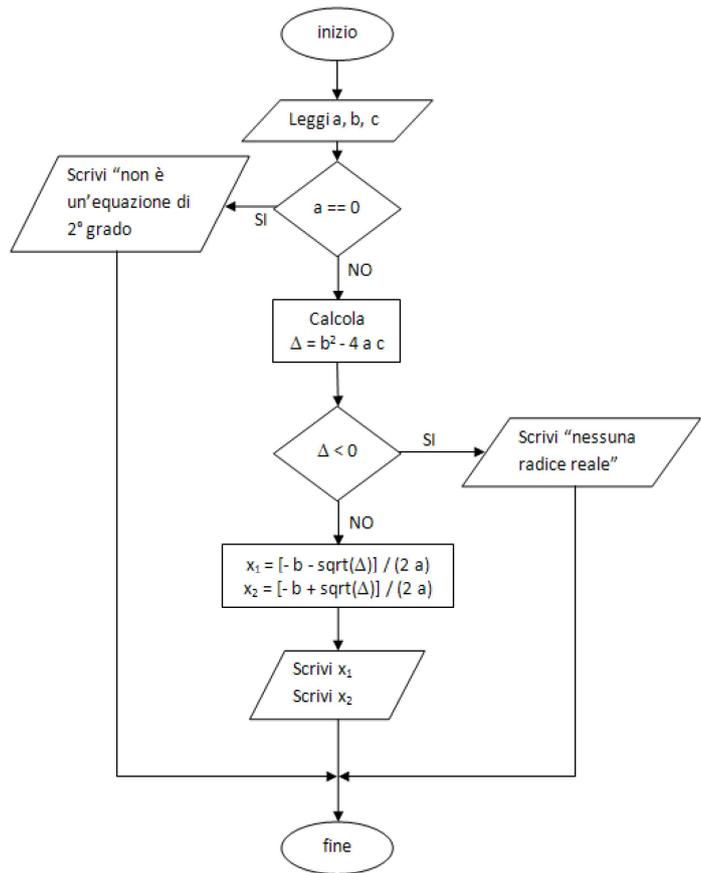
**effettivo** cioè ho sempre lo stesso risultato se inserisco gli stessi valori di a, b e c

**definito** cioè vale per qualunque valore di Delta

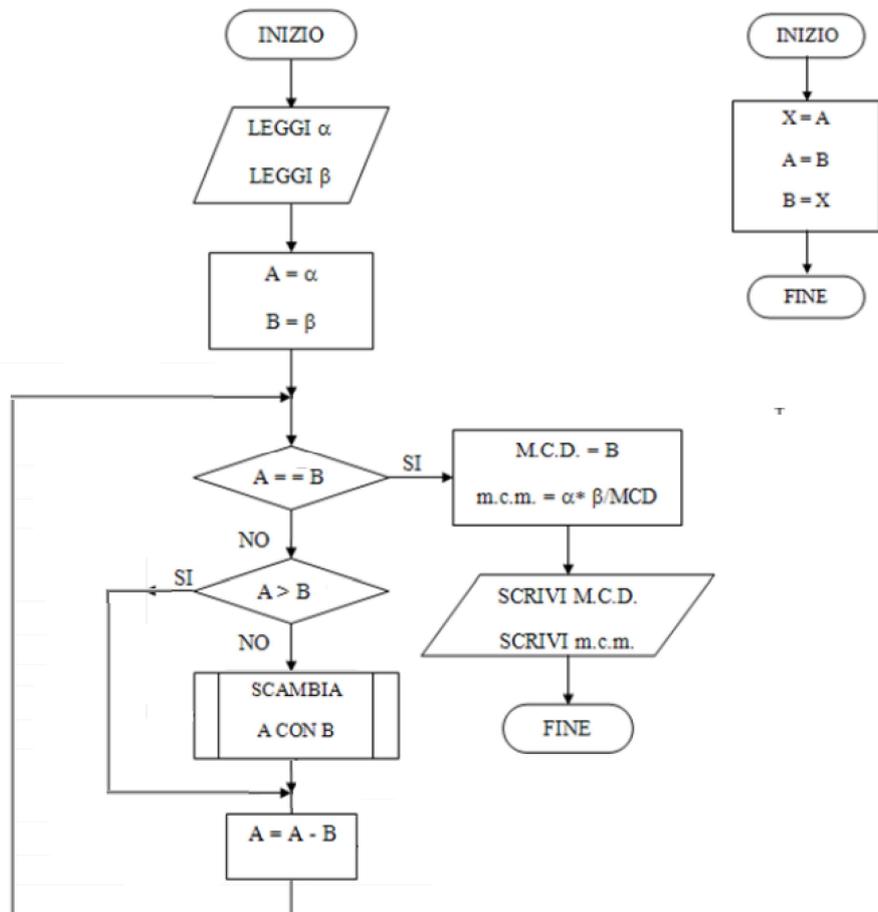
**non ambiguo** cioè il percorso tra inizio e fine è interpretato da tutti allo stesso modo

**generale** cioè vale per qualunque valore di a, b e c

**finito** ho in tutto 10 blocchi



## ALGORITMO DI EULERO



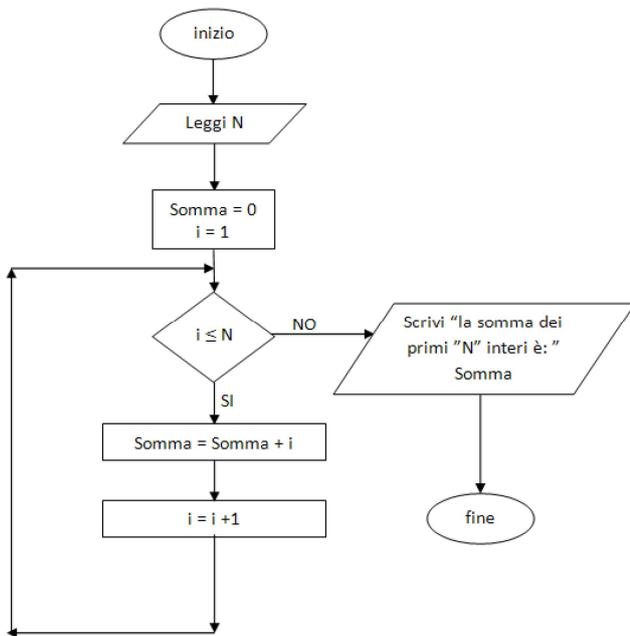
## I CICLI (BLOCCHI DI ITERAZIONE STRUTTURATA)

Sono combinazioni di blocchi elementari che si presentano sempre con la stessa struttura, nei quali ci sono alcuni blocchi vengono eseguiti più volte

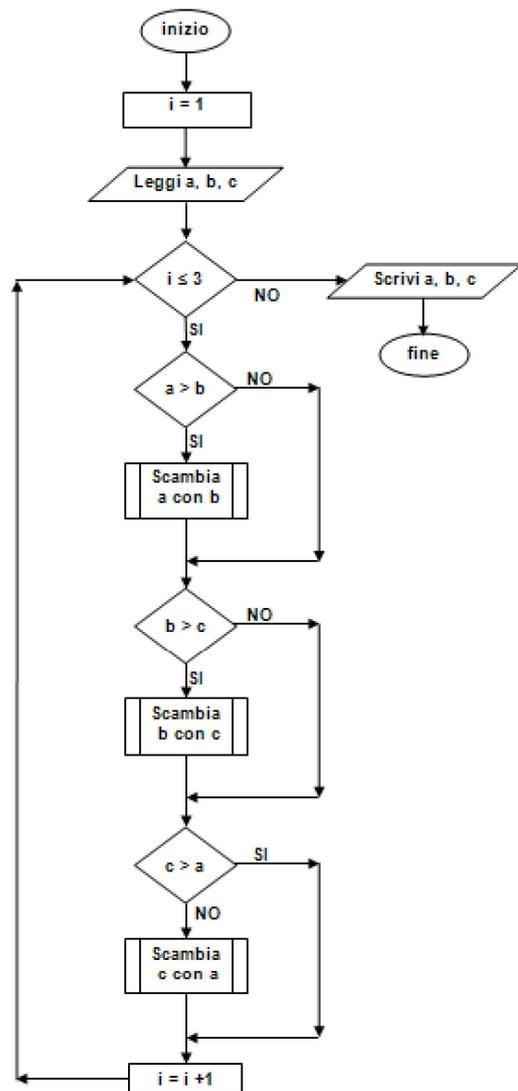
**Si suddividono in:**

- **blocchi di iterazione definita:** conosco a priori prima di entrare nel ciclo il numero di volte che verrà ripetuto il ciclo stesso
- **blocchi di iterazione indefinita:** non conosco a priori il numero di volte che verrà ripetuto il ciclo, tale numero dipende dal risultato della verifica eseguita nel blocco condizionale

definito	indefiniti	
FOR	WHILE	DO WHILE
<p>ciclo <b>FOR</b></p> <p>la variabile N indica quante volte deve essere ripetuto il ciclo</p> <p>i (indice o contatore) serve a tenere il conto di quante volte è già stato ripetuto il ciclo, ha il valore iniziale uguale a 0 o 1, viene incrementato generalmente di 1</p>	<p>ciclo <b>WHILE</b></p> <p>si verifica come prima cosa la condizione iniziale, se questa è falsa non si esegue mai il ciclo.</p> <p>se la condizione è inizialmente vera, il ciclo viene ripetuto un numero non definito di volte fintantoché la condizione rimane vera.</p>	<p>ciclo <b>DO - WHILE</b></p> <p>si esegue per prima cosa l'espressione (che viene quindi eseguita almeno una volta)</p> <p>poi si verifica la condizione, se la condizione è falsa si esce dal ciclo</p> <p>se la condizione è vera il ciclo viene ripetuto un numero non definito di volte fintantoché la condizione rimane vera.</p>
<p>N deriva dalla parte superiore dell'algoritmo</p> <pre> graph TD     Start(( )) --&gt; I1[i = 1]     I1 --&gt; Cond{i &lt;= N}     Cond -- FALSO --&gt; Exit(( ))     Cond -- VERO --&gt; Calc[calcoli qualunque essi siano]     Calc --&gt; Inc[i = i + 1]     Inc --&gt; Cond     </pre>	<pre> graph TD     Start(( )) --&gt; Cond{CONDIZIONE}     Cond -- FALSO --&gt; Exit(( ))     Cond -- VERO --&gt; Calc[calcoli qualunque essi siano]     Calc --&gt; Cond     </pre>	<pre> graph TD     Start(( )) --&gt; Calc[calcoli qualunque essi siano]     Calc --&gt; Cond{CONDIZIONE}     Cond -- FALSO --&gt; Exit(( ))     Cond -- VERO --&gt; Calc     </pre>



algoritmo della somma dei primi N numeri naturali



algoritmo di ordinamento crescente

## GLI PSEDUDOLINGUAGGI

In informatica per pseudolinguaggio si intende un linguaggio di programmazione fittizio, non direttamente eseguibile da un computer

Lo pseudolinguaggio può essere utilizzato alternativamente al diagramma di flusso e non è soggetto a molte limitazioni intrinseche di quest'ultimo tipo di rappresentazione; non esiste uno pseudolinguaggio standard e convenzionalmente usato da tutti, ciascun programmatore può essere portato ad utilizzare una propria variante.

### Pseudolinguaggio dell'algoritmo di Eulero

Inizio - Leggi dalla tastiera  $\alpha$  e  $\beta$  - Assegna ad A il valore di  $\alpha$  - Assegna a B il valore di  $\beta$

(\*) Se il valore di A è uguale al valore di B Assegna al M.C.D. il valore di B - Calcola il valore del m.c.m. come prodotto del valore di  $\alpha$  e  $\beta$  diviso il valore del M.C.D. - Scrivi sul monitor il valore del M.C.D.e il valore del m.c.m. - FINE

Altrimenti Se il valore di A non è uguale al valore di B - se il valore di A è più grande del valore di B - Allora scambia i valori di A e B - fai la differenza tra il valore di A e il valore di B - Vai alla riga (\*)

Esercizi:

- 1) Determinare il massimo di tre numeri.
- 2) Determinare perimetro ed area di un triangolo scaleno (usando la formula di Erone).
- 3) Calcolare il perimetro e l'area di un poligono regolare (conoscendo l'apotema in funzione dei lati).
- 4) Calcolo del fattoriale di un numero ( $6! = 6 * 5 * 4 * 3 * 2 * 1$ ).
- 5) Calcolare la media aritmetica di un insieme di numeri.
- 6) Calcolare la radice di un'equazione di primo grado.
- 7) Calcolare le radici di un'equazione di secondo grado.
- 8) Sul prezzo di un prodotto viene praticato lo sconto del 3% se costa meno di 1.000 €, e del 5% se ne costa di più. Inserito da tastiera il prezzo P, calcolare il prezzo da pagare secondo la regola sopra descritta.
- 9) Sul prezzo di un biglietto di un treno viene applicato un supplemento del 7% se il treno è di tipo "a", del 12% se è di tipo "b" e del 18% se è di tipo "c".  
Per gli altri treni non c'è supplemento. Calcolare il prezzo totale del biglietto, a seconda del tipo di treno e comunicare il tipo di treno con il prezzo calcolato.
- 10) Dati in input 6 valori dare in output separatamente la somma dei numeri positivi e la somma dei negativi.
- 11) Calcolare la somma dei primi numeri N naturali e pari.
- 12) Calcolare la somma dei primi numeri N naturali e multipli di 3.
- 13) Dato in input secondi dare in output i min, ore e sec.
- 14) Dato in input una parola dare in output il numero delle vocali.
- 15) Dato in input un valore decimale trasformarlo in binario.
- 16) Dato in input un numero intero scrivere tutti i divisori del numero e contarli.
- 17) Supponiamo di avere due bicchieri, denominati A e B, pieni rispettivamente uno di acqua e l'altro di vino, e di voler travasare A in B e viceversa.
- 18) Un pastore deve attraversare un fiume portando sull'altra riva un lupo e una capra affamati e una cassa di cavoli. Ha a disposizione una barca a remi con la quale può traghettare un solo oggetto o animale alla volta. Ma, attenzione! Non può lasciare da soli: il lupo e la capra perché il lupo si mangia la capra; la capra ed i cavoli perché la capra si mangia i cavoli.
- 19) Un anno è bisestile se il suo numero è divisibile per 4, con l'eccezione che gli anni secolari (quelli divisibili per 100) che sono bisestili solo se divisibili per 400. Sono cioè bisestili tutti gli anni la cui numerazione termina con le due cifre 04, 08, 12... fino a 96; gli anni che terminano con 00 sono bisestili solo se l'anno è divisibile per 400, cioè il 1600, il 2000, il 2400 eccetera.
- 20) Calcolare la potenza di un numero procedendo per moltiplicazioni successive.