
Dal Problema all'Algoritmo

Prof. Francesco Accarino
IIS Altiero Spinelli Sesto San Giovanni

Cosa Impareremo

- Informazioni e codici
- Definizione e soluzione di un problema
- Individuazione dei dati
- Definizione e caratteristiche degli Algoritmi
- Rappresentazione degli Algoritmi (Flow Chart)
- Programmazione Strutturata

Informatica e Informazione

L' Informazione è tutto ciò che:

- ❑ Ha un significato o un'utilità
- ❑ Può essere comunicata (scambiata) tra due soggetti anche con modalità diverse (scritta, orale grafica ..)
- ❑ Può assumere forme diverse (simboli, testi, numeri, grafici, disegni, Immagini, colori...)
- ❑ Può essere memorizzata
- ❑ Può essere elaborata e trasformata nella forma più utile a seconda della necessità



Varie Forme di Informazioni

numeriche 1.243.740

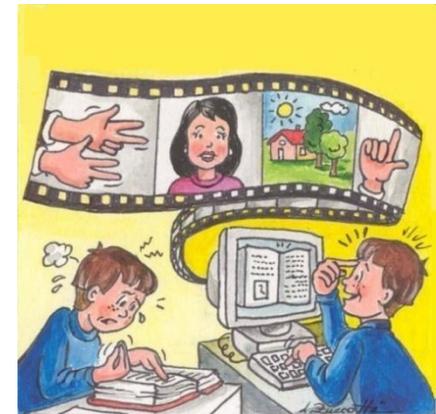
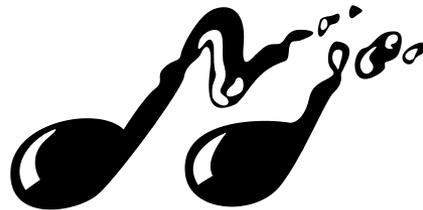
alfabetiche **Azzurra Rossi**

alfanumeriche **Via Verdi, 17 Porto Azzurro**



iconiche e grafiche

sonore



L'uomo ha sempre cercato sistemi per:

I
N
F
O
R
M
A
Z
I
O
N
I

I
N
F
O
R
M
A
Z
I
O
N
I

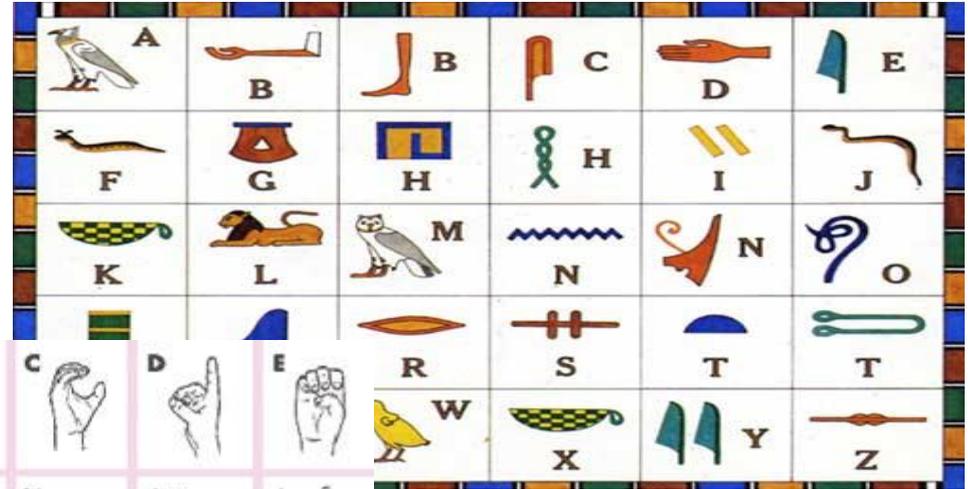
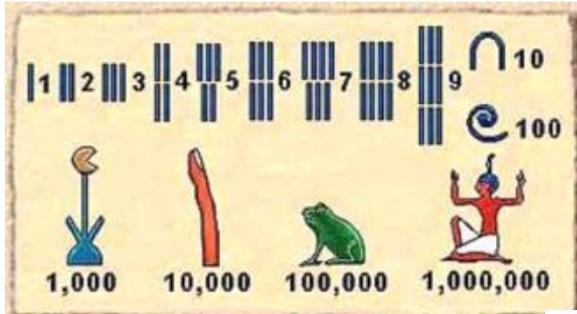
Memorizzare

Scambiare



Elaborare

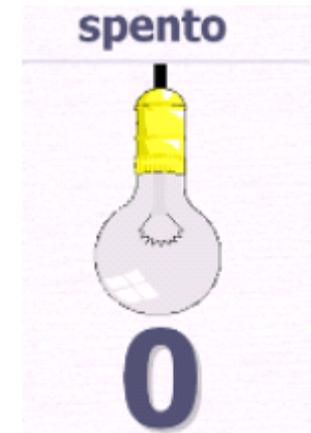
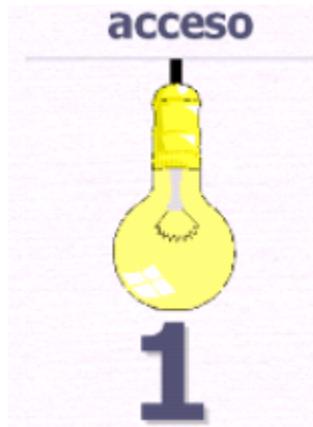
Per rappresentare le Informazioni Bisogna Inventare dei Simboli e attribuirgli un significato. (CODICI)



1	I	30	XXX
2	II	40	XL
3	III	50	L
4	IIII (IV)	60	LX
5	V	70	LXX
6	VI	80	LXXX
7	VII	90	XC
8	VIII	100	C
9	IX (VIII)	200	CC
10	X	300	CCC
11	XI	400	CD
12	XII	500	D (IO)
13	XIII	600	DC
14	XIV	700	DCC
15	XV	800	DCCC
16	XVI	900	CM
17	XVII	1.000	M (CIO)
18	XVIII	2.000	MM, CIOCIO, II
19	XIX	10.000	CCICOO, X̄
20	XX	100.000	CCCCICOO, C̄
		1.000.000	CCCCICOOO, X̄



Informatica



Infomazione+auto**matica**

Risolvere Problemi

Uno degli scopi fondamentali dell'informatica è:

la risoluzione di problemi in modo automatico

problema = compito che si vuole far risolvere automaticamente ad un **Esecutore**



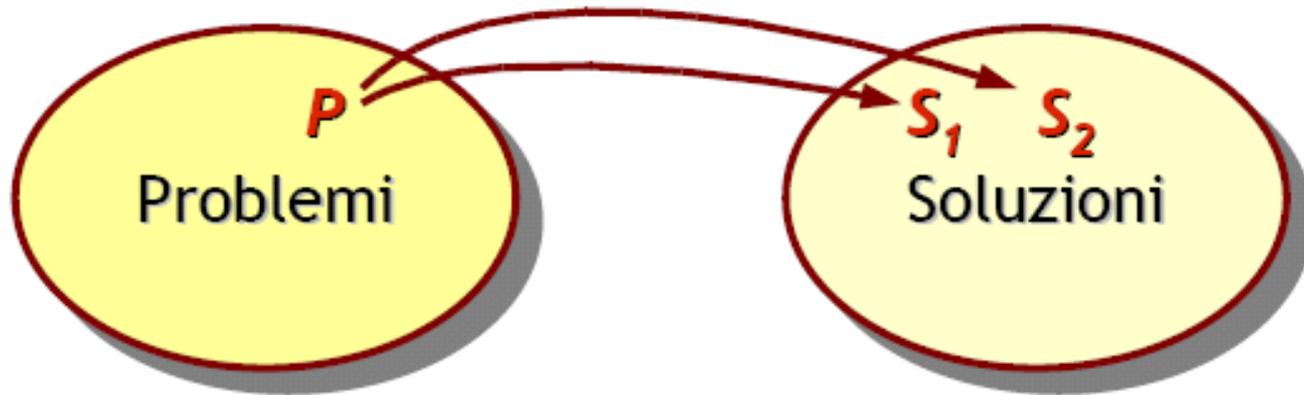
Definizione di Problema

■ **Problema**

*Quesito con cui si chiede di trovare,
mediante un procedimento di calcolo,
uno o più dati sconosciuti,
partendo dai dati noti*

Soluzione di un problema

- **Insieme finito di attività** da compiere per ottenere un *effetto* o un *risultato* desiderato



Esempio Moltiplicazione per raddoppio

impiegata nell'antico Egitto

Esempio: 18 per 13 = 234

si compila una tabella a due colonne la prima riga è costituita da 1 e da uno dei due fattori;

1	18
---	----

finché nella prima colonna si ottengono numeri non maggiori del secondo fattore si fanno seguire altre righe ottenute raddoppiando gli elementi della riga precedente

1	→	18
---	---	----

2		36
---	--	----

4	→	72
---	---	----

8	→	144
---	---	-----

16		288
---------------	--	----------------

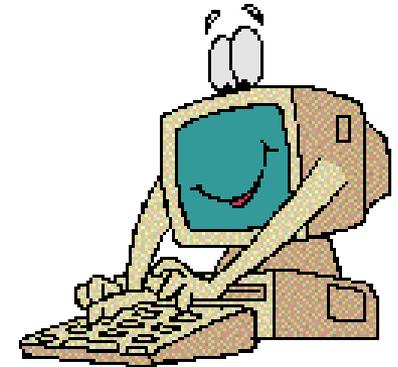
Si scelgono gli elementi della prima colonna di somma pari al secondo fattore (13) la somma dei corrispondenti elementi della seconda colonna fornisce il risultato:

$$1+4+8 = 13 \text{ quindi: } 18+72+144 = 234$$

Soluzione di un problema

- Passaggio dalla formulazione del problema all'individuazione del metodo solutivo
 - Analisi del testo e individuazione dei dati
 - Ricerca del metodo di soluzione
 - Esecuzione

L'esecuzione può essere delegata ad un esecutore che capisca la descrizione della soluzione cioè sia in grado di eseguire le operazioni richieste



Risoluzione di Problemi

E' molto importante acquisire una metodologia di lavoro

- **Individuazione e rappresentazione dei dati**
 - Quali dati sono coinvolti
 - Quale soluzione ci si aspetta
 - Scartare i dati inutili e concentrarsi su quelli utili
 - Individuare eventuali vincoli o aggiungerne se ritenuti opportuni

Individuazione e rappresentazione dei DATI

I dati possono essere distinti in base a :

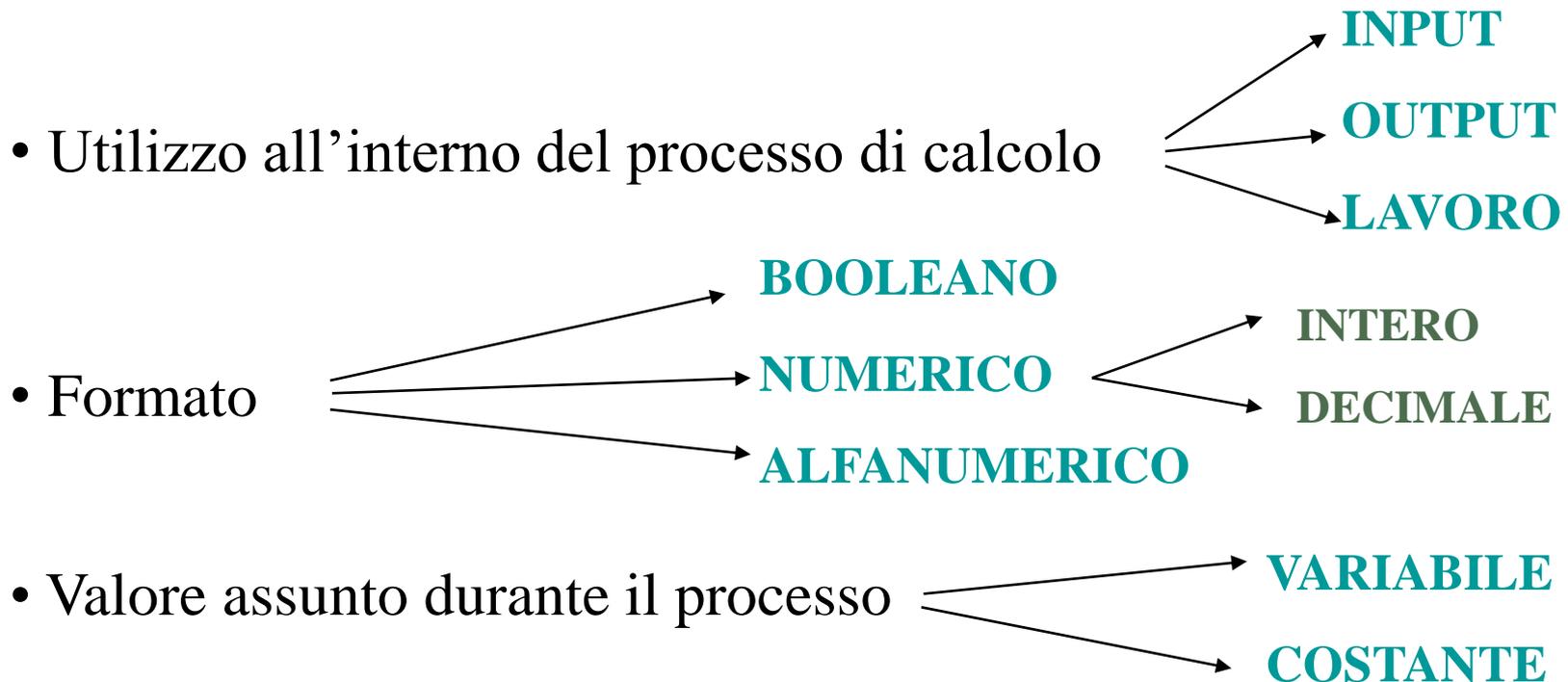


Tabella dei dati

Identificatore	Input output lavoro	Variabile costante	Descrizione	Tipo
Nome1	input	Variabile	Operando....	Numerico Intero senza segno
Nome2	input	Variabile	Attributo...	Alfanumerico
Nome3	Output	Variabile	Risultato....	Booleano
Nome4	Lavoro	Variabile	Calcolo.....	Numerico decimale con segno
Nome5	Lavoro	Costante	Calcolo.....	Numerico decimale senza segno

Ricerca del metodo di soluzione

La soluzione del problema deve essere ricercata utilizzando le capacità del risolutore e rappresenta:

La serie di passi o istruzioni che costituiscono l'insieme delle azioni da compiere per poter risolvere un problema.

Viene chiamata **Algoritmo** dal nome del Matematico Arabo:

Abu Ja'far Muhammad ibn Musa Al-Khwarizmi

Che per primo formalizzò i metodi per eseguire le operazioni aritmetiche

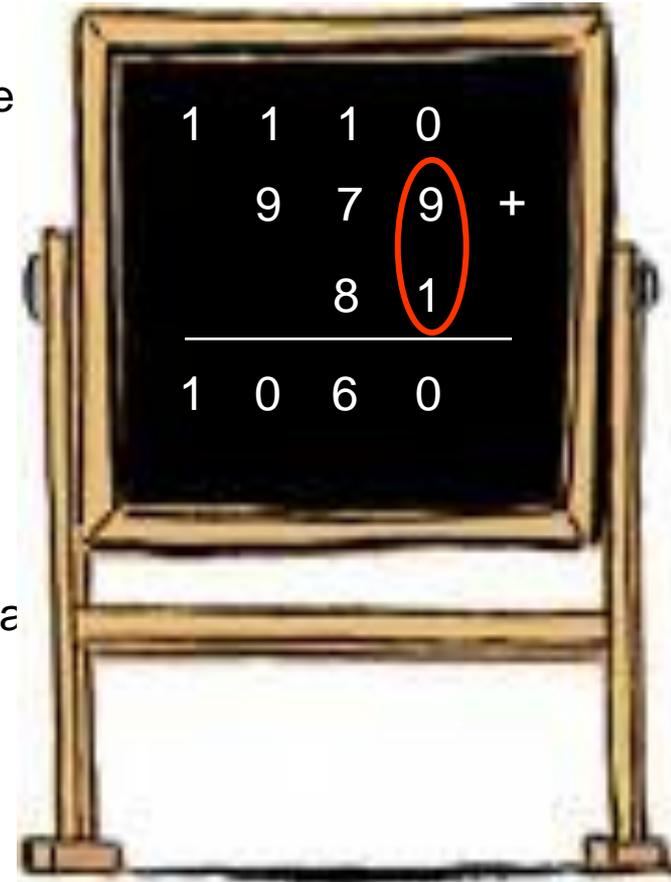


Somma di due numeri a mano

- **problema** : Effettuare la somma tra due numeri rappresentati in cifre decimali su una Lavagna
- L'esecutore è in grado di comprendere il significato delle cifre decimali, di leggere e scrivere cifre su una lavagna, e di calcolare la somma e il riporto della somma di due cifre decimali e di un riporto

Algoritmo di Al-Khwarizmi

- Si *inizializzi* la lavagna: si pulisca la lavagna e si scrivano i due numeri uno sotto l'altro incolonnati a destra
- n *Passo 1*: si consideri la coppia costituita dalle cifre più a destra dei due numeri e si consideri come riporto iniziale 0
- n *Passo 2*: si calcoli la somma e il riporto della coppia di cifre e del riporto considerati e si scriva la somma sotto le due cifre considerate
- n *Passo 3*: Finché vi sono cifre a sinistra di quelle appena considerate, si ripeta il passo 2 considerando la coppia costituita dalle cifre dei due numeri immediatamente a sinistra e il riporto appena calcolato.
- n *Passo 4*: Se l'ultimo riporto calcolato è diverso da 0, scriverlo a sinistra dell'ultima somma scritta sulla lavagna.
Fine: il risultato è scritto sulla lavagna sotto i due addendi



Algoritmo

Definizione formale

sequenza di azioni, valida per un insieme di dati iniziali ben definito, che, compiuta da un esecutore, trasformi i dati nel risultato finale, attraverso un numero finito di passi elementari e non ambigui.

Un algoritmo deve presentare un punto di **INIZIO** (dove comincia il procedimento risolutivo) e un punto di **FINE** (raggiunto il quale si interrompe l'esecuzione delle azioni)

Algoritmo

Caratteristiche

- **completo ed esaustivo**

per tutti i casi che si possono verificare durante l'esecuzione, deve essere indicata la soluzione da seguire

- **riproducibile**

ogni successiva esecuzione dello stesso algoritmo con i medesimi dati iniziali deve produrre sempre i medesimi risultati finali
deve risolvere medesime categorie di problemi

- **deterministico**

le azioni di cui è composto devono fornire la medesima, certa soluzione, indipendentemente dalla natura dell'esecutore

Le azioni descritte nell'algoritmo devono essere:

- in numero finito
- non ambigue
- realizzabili

Esempi intuitivi di algoritmi

- Le azioni necessarie per l'utilizzo di un elettrodomestico
- le regole da seguire per la divisione di 2 numeri interi
- le indicazioni per la consultazione di un vocabolario

Algoritmo: gli elementi coinvolti

Nella definizione si individuano tre elementi fondamentali:

Dati:

Iniziali (o in ingresso)

sono gli elementi che vengono elaborati dall'algoritmo

Finali (o in uscita)

sono i risultati prodotti dall'algoritmo

•Sequenza di azioni (istruzioni, passi elementari)

un'azione è un'operazione elementare (istruzione) che compone un passo della serie di operazioni che deve essere eseguita sui dati di ingresso per ottenere il risultato, cioè i dati in uscita

•Esecutore (o processore)

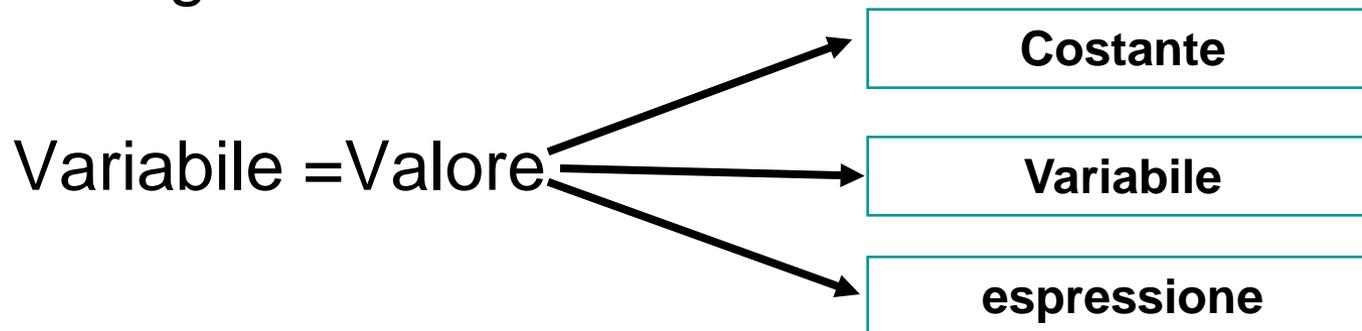
è il soggetto che compie le azioni, cioè legge le istruzioni che devono essere eseguite sui dati in ingresso, le interpreta e le esegue in modo da elaborare tali dati per trasformarli in risultati, cioè dati in uscita. l'insieme delle istruzioni deve essere scelto a seconda dell'esecutore, in termini di qualità, di tipologia e di livello di dettaglio

Considerazioni sull'algoritmo della Somma

Si consideri come riporto 0 : $\longrightarrow R=0$

Si calcoli la somma e il riporto della coppia di cifre e del riporto considerati e si scriva la somma sotto le due cifre considerate: $\longrightarrow S=9+1+0$

Un'azione fondamentale è quella detta di Assegnamento:



Espressione:

- Un espressione può essere costruita combinando gli **operandi** mediante **operatori**
- Operandi: Costanti Variabili o espressioni
- Operatori: Aritmetici di relazione o logici

Operatori aritmetici	
+	addizione
-	sottrazione
*	moltiplicazione
/	Divisione
%	Calcolo del resto della divisione tra interi

Espressione:

- Un espressione può essere costruita combinando gli **operandi** mediante **operatori**
- Operandi: Costanti Variabili o espressioni
- Operatori: Aritmetici di relazione o logici



Operatori di relazione	
<	Minore di
<=	Minore o uguale di
>	Maggiore di
>=	Maggiore o uguale di
==	Uguale
!=	Diverso

Espressione:

- Un espressione può essere costruita combinando gli **operandi** mediante **operatori**
- Operandi: Costanti Variabili o espressioni
- Operatori: Aritmetici di relazione o logici



Operatori logici		
And	&&	Per il prodotto logico (congiunzione)
Or	 	Per la somma logica (disgiunzione)
Not	!	Per la negazione

Espressioni

Un espressione Aritmetica può essere ad esempio: $((A*B) + (C/3)) - (D\%2)$

Si utilizzato gli operatori aritmetici e le parentesi per indicare la priorità delle operazioni

Un espressione Logica invece è espressa legando tra di loro più predicati logici.

Un predicato logico è una relazione tra due variabili o tra una variabile e una costante. **Si utilizzano gli operatori di relazione.**

Esempio: $A > B$; $A == C$ $A >= 5$; ecc. il loro valore può essere **vero** o **falso**

E quindi un'espressione logica può essere ad esempio: $(A > B) \&\& (A == C)$

1 0

Il valore di un'espressione Logica è comunque **vero** o **falso**

Espressioni Logiche

Per Valutare le espressioni logiche si utilizzano le cosiddette tavole di verità

Dati due predicati A e B

AND		
A	B	A and B
Falso	Falso	Falso
Vero	Falso	Falso
Falso	Vero	Falso
Vero	Vero	Vero

NOT	
A	Not A
Falso	Vero
Vero	Falso

OR		
A	B	A or B
Falso	Falso	Falso
Vero	Falso	Vero
Falso	Vero	Vero
Vero	Vero	Vero

A=7; B=5; C=7;

(A>B)&&(A==C)

VERO

Altre Considerazioni sull'algoritmo della Somma

3: Finché vi sono cifre a sinistra di quelle appena considerate, si ripeta il passo 2 

Un'azione fondamentale in un algoritmo è l'individuazione di un certo numero di passi da ripetere più volte finché una certa condizione è verificata .

Questa azione si chiama: **Iterazione**

4: Se l'ultimo riporto calcolato è diverso da 0, scriverlo a sinistra dell'ultima somma scritta sulla lavagna. 

Ultima considerazione è la possibilità di eseguire operazioni solo se certe condizioni sono verificate

Questa azione si chiama: **Scelta**

Rappresentazione degli Algoritmi

Una volta individuati i passi che costituiscono l'algoritmo è utile rappresentarlo con un modello chiamato Flow Chart (Diagramma di Flusso)

E' un modello Grafico ed utilizza i seguenti simboli:



Inizio e fine



Input e Output



Assegnamento



Condizione

Come costruire Flow Chart Ben strutturati

- Un flow chart deve essere chiaro e lineare e per ottenere ciò bisogna attenersi alle regole dettate dal:

Teorema di Bohm – Jacopini

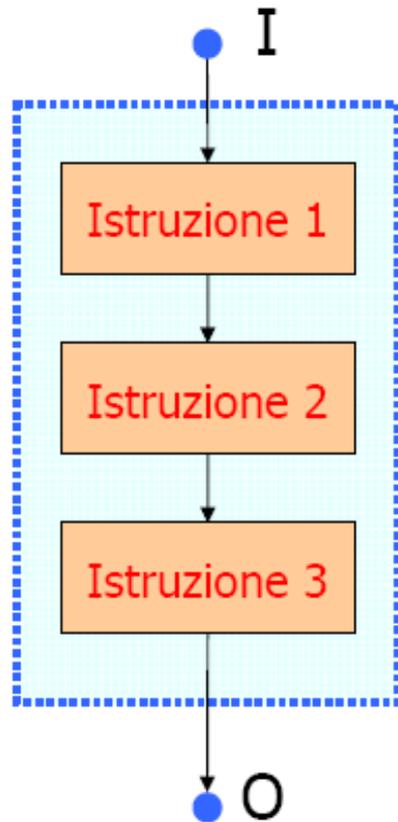
(In versione semplificata)

Le strutture di **sequenza selezione e iterazione** sono sufficienti ad esprimere **qualsiasi algoritmo**

Ogni struttura deve avere un unico punto di inizio ed un unico punto di fine. **Soprattutto non devono incrociarsi ed accavallarsi**

Struttura di sequenza o BLOCCO

Diagramma a blocchi



Pseudo-codice

```
{  
  Istruzione 1  
  Istruzione 2  
  Istruzione 3  
}
```

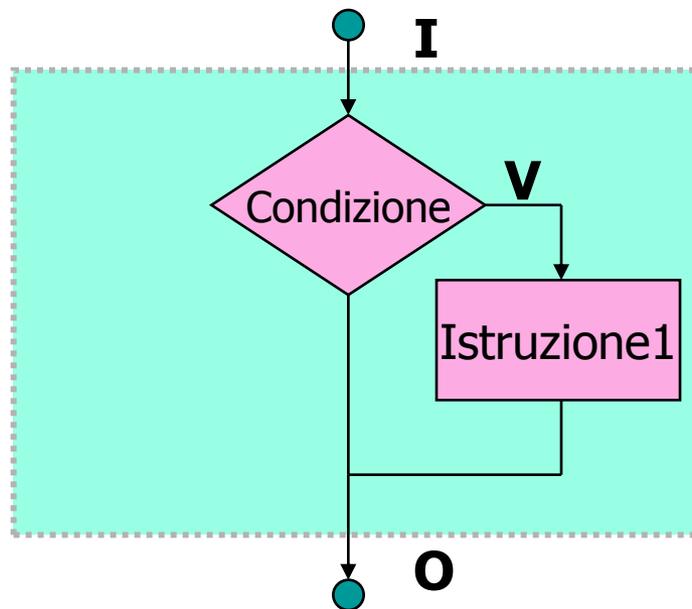
Le tre istruzioni vengono eseguite una di seguito all'altra.

IL concetto di blocco permette di considerare la sequenza di più istruzioni come un'unica istruzione composta

Struttura condizionale a una via

Diagramma a blocchi

Pseudo-codice

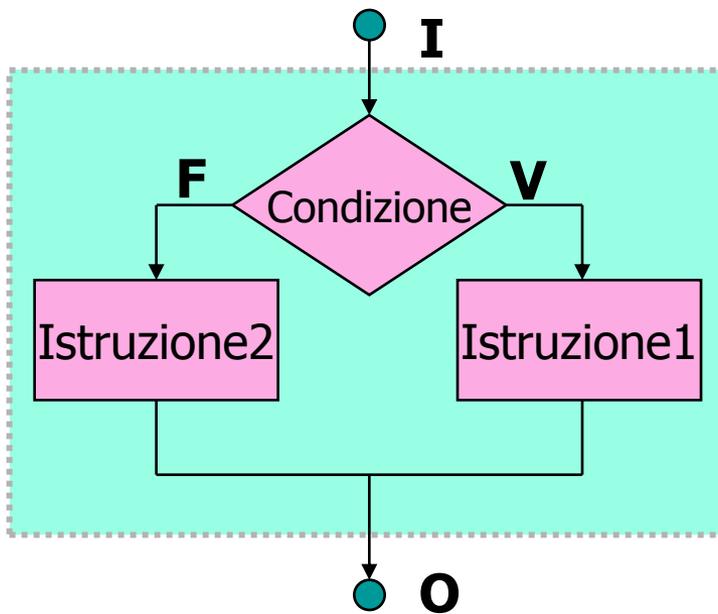


if condizione
istruzione

SE la **condizione** è vera viene eseguita **Istruzione1**
altrimenti si prosegue

Struttura condizionale (doppia scelta)

Diagramma a blocchi



Pseudo-codice

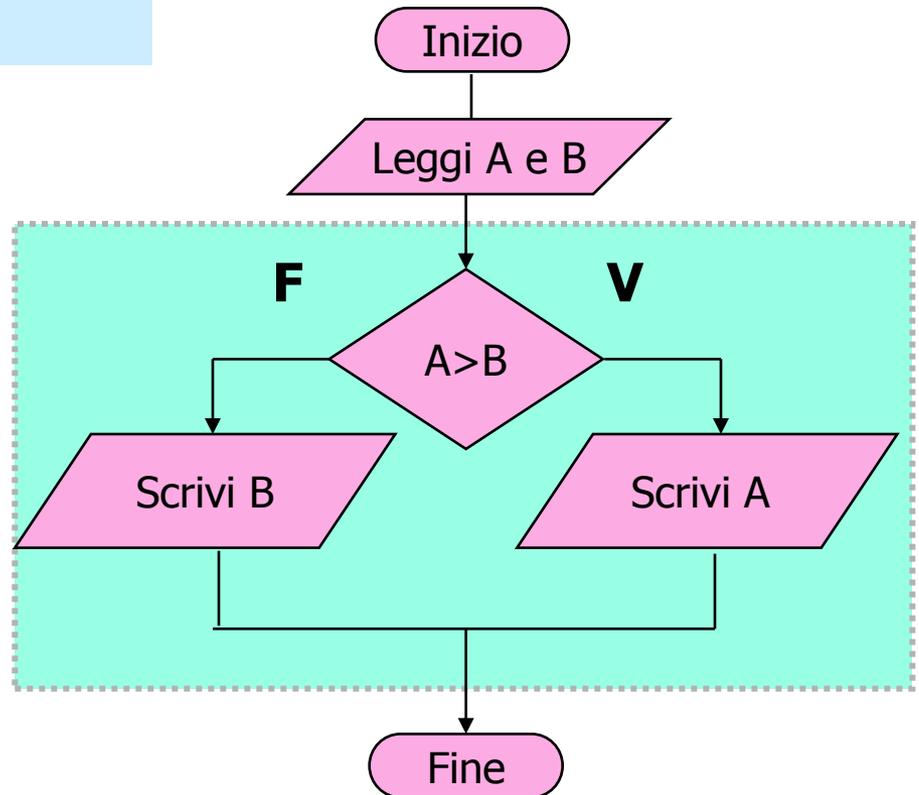
```
if condizione  
    istruzione 1  
else  
    istruzione 2
```

SE la **condizione** è vera viene eseguita **Istruzione1** e si prosegue
altrimenti viene eseguita **Istruzione2** e si prosegue

Struttura condizionale (Esempi)

Es. Determina il più grande tra due numeri inseriti da tastiera.

```
leggi A,B;  
if (A > B)  
    scrivi A;  
else  
    scrivi B;
```

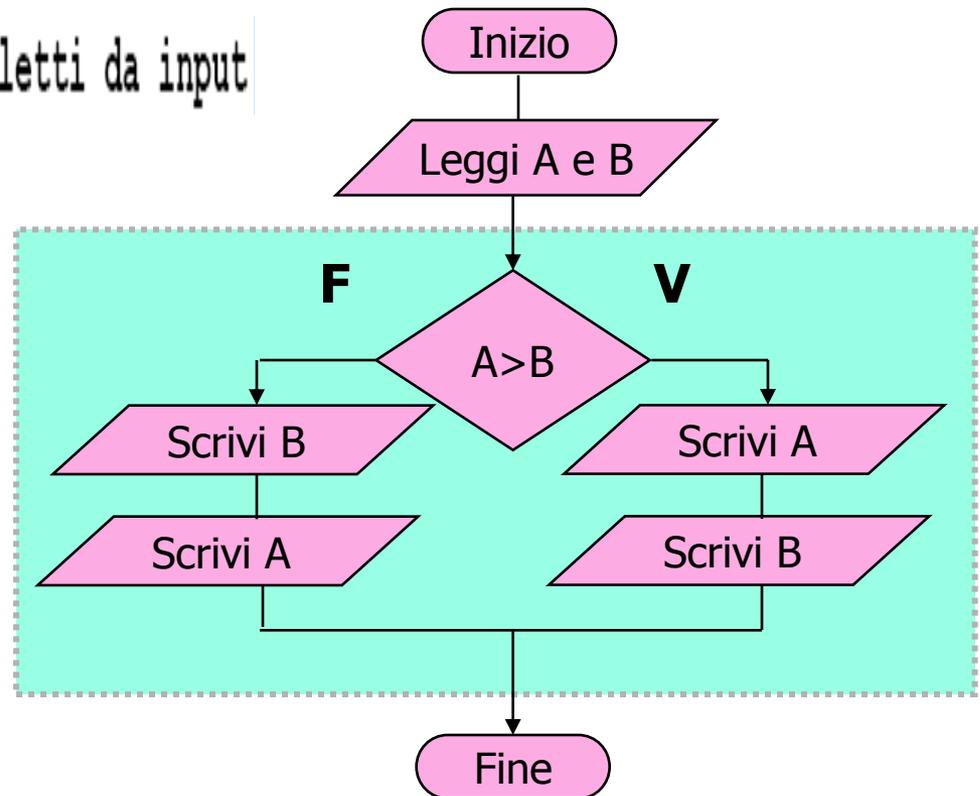


Struttura condizionale (Esempi)

- `<istruzione1>` e `<istruzione2>` sono ciascuna una *singola istruzione*
- Qualora in un ramo occorra specificare più istruzioni, si deve quindi utilizzare un *blocco*

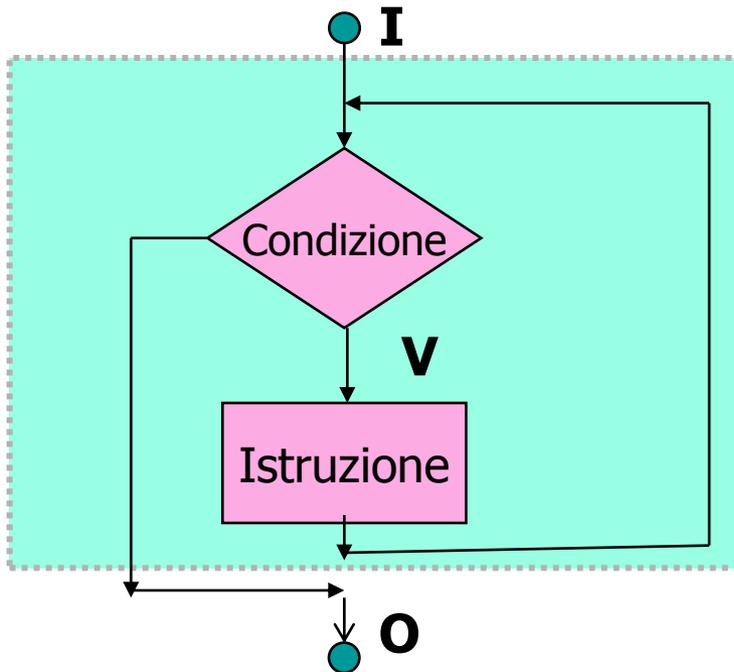
Es: scrivi in ordine crescente due numeri letti da input

```
leggi A, B;  
if (A < B) {  
    scrivi A;  
    scrivi B;  
}  
else {  
    scrivi B;  
    scrivi A;  
}
```



Struttura di iterazione precondizionata

Diagramma a blocchi



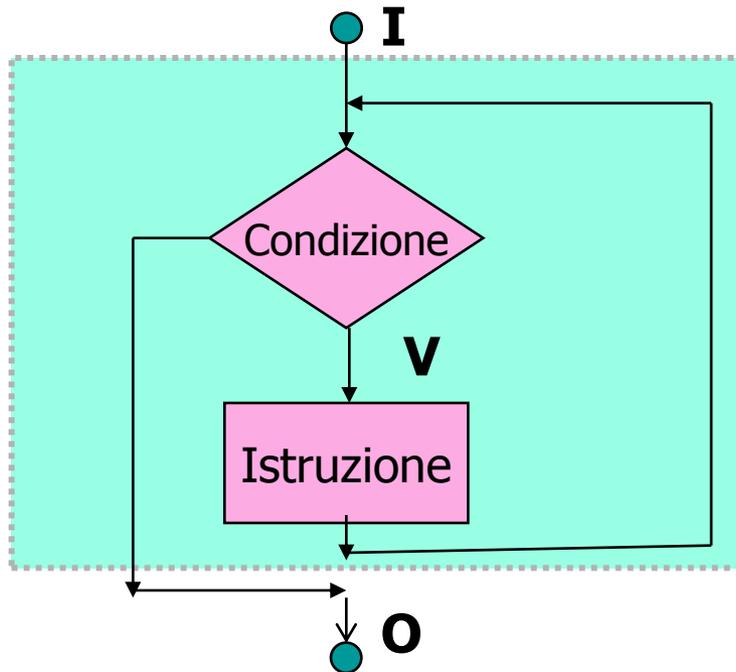
Pseudo-codice

```
while condizione  
  istruzione
```

Mentre la **condizione** è vera viene eseguita **Istruzione**
Appena è falsa si prosegue

Struttura di iterazione preconditionata

while condizione
istruzione

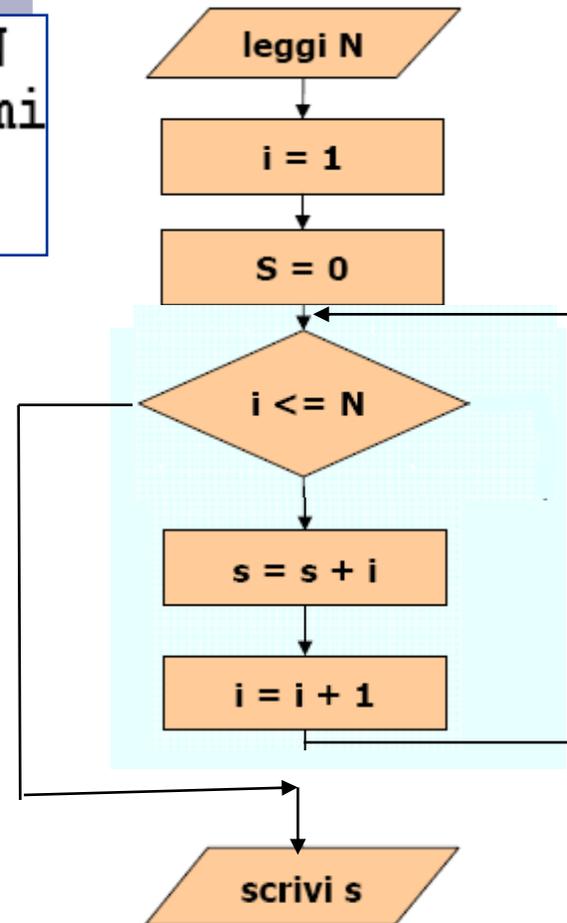


- **<istruzione>** viene ripetuta *per tutto il tempo in cui la condizione rimane vera*
- Se la condizione è già inizialmente falsa, l'iterazione non viene eseguita *neppure una volta*
- In generale, *non è noto a priori quante volte* l'istruzione sarà ripetuta

Struttura di iterazione preconditionata

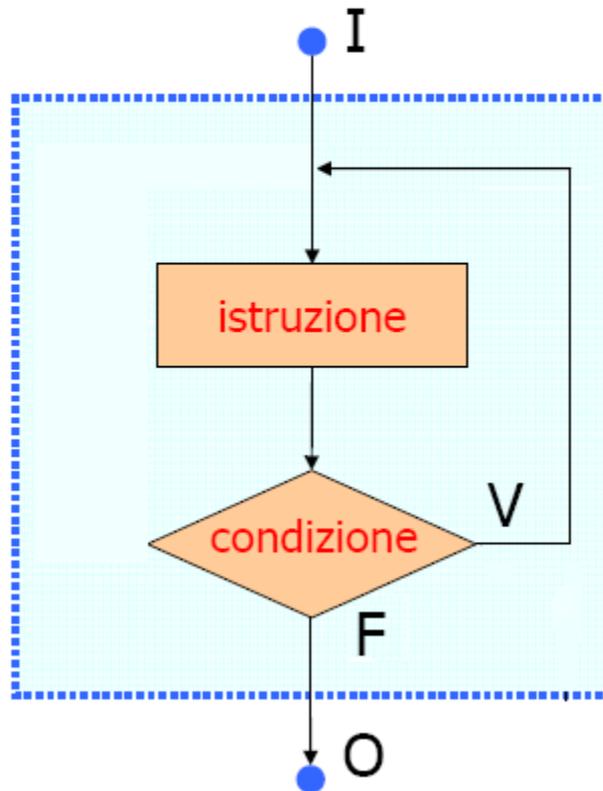
Es: Letto un numero intero N da input, sommare i primi N numeri positivi e scrivere il risultato

```
leggi N;  
i=1;  
s=0;  
while (i<=N) {  
    s = s + i;  
    i = i + 1;  
}  
scrivi s;
```



Struttura di Iterazione Postcondizionata

Diagramma a blocchi



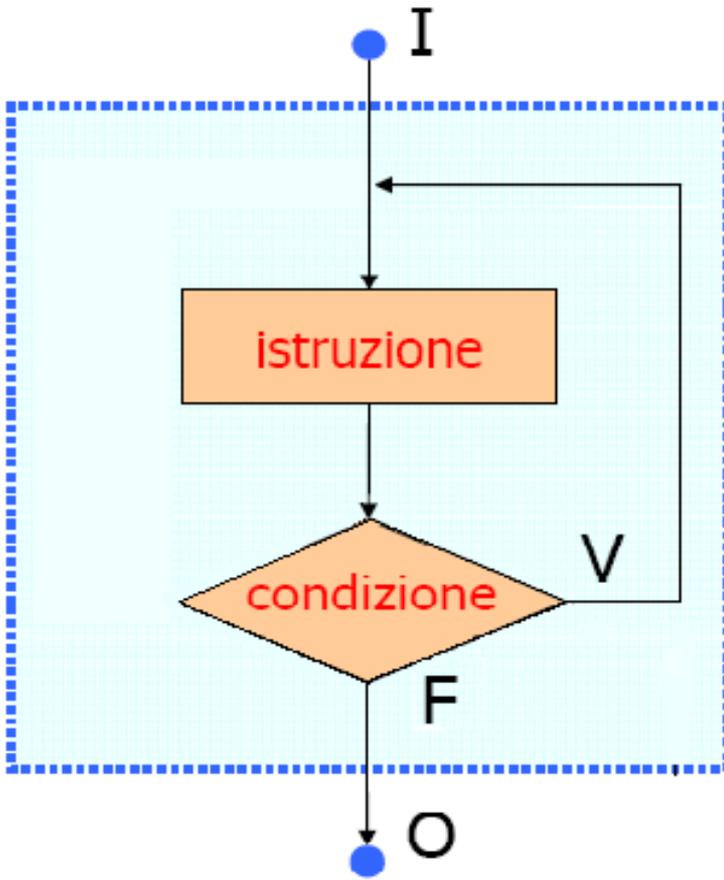
Pseudo-codice

```
do  
istruzione  
while condizione
```

```
Esegui  
Istruzione  
Finche la condizione è vera
```

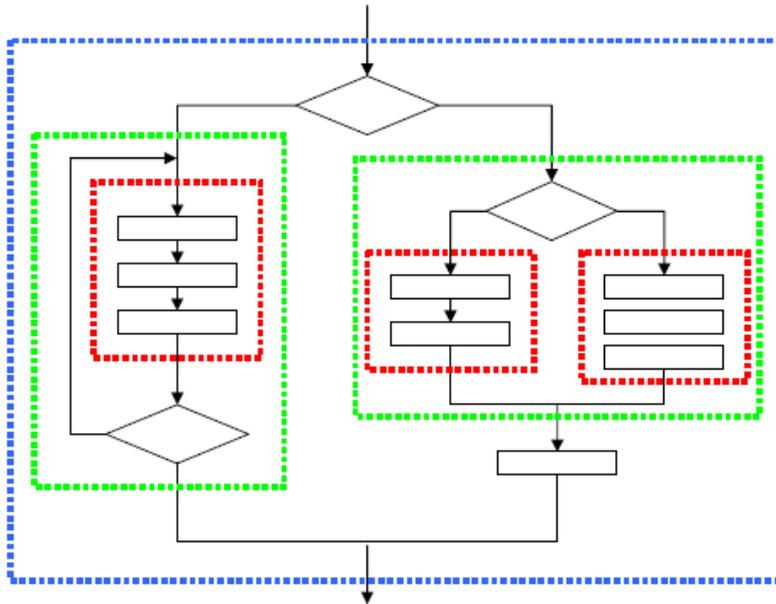
Struttura di iterazione Postcondizionata

do
istruzione
while condizione

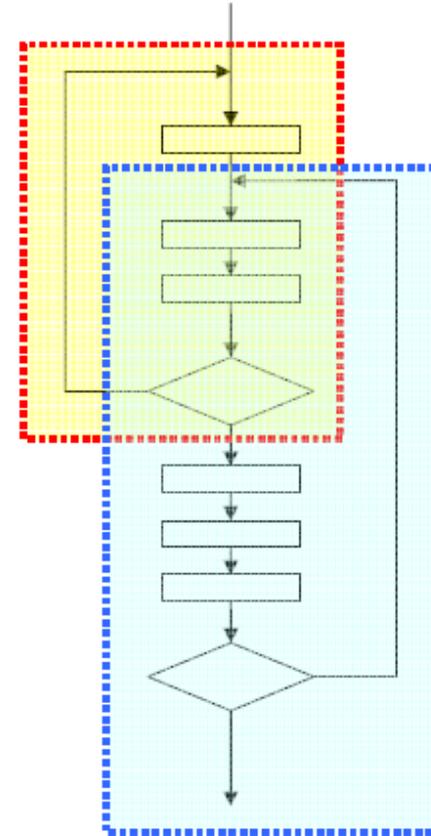


- È una variante della precedente: la condizione viene verificata **dopo** aver eseguito **<istruzione>**
- Se la condizione è falsa, l'istruzione **viene comunque eseguita almeno una volta**

Schemi a blocchi strutturati



Corretto



**Errato:
Spaghetti Programming**

Conclusioni:

Con la stesura del Flow Chart termina la fase di analisi del problema i cui passi sono:

- Interpretazione del testo e individuazione dei dati
- Rappresentazione dei dati assegnando identificatori, uso, e tipi: **Tabella**
- Ricerca della soluzione come sequenza di passi in forma discorsiva
- rappresentazione dell'algoritmo con il flow chart
- Verifica della soluzione adottata.

Le altre Fasi sono:

- Programmazione cioè la traduzione dell'algoritmo in un linguaggio di programmazione
- Implementazione cioè scegliere un adeguato sistema di calcolo e un adeguato sistema di sviluppo per creare l'eseguibile cioè il programma vero e proprio.