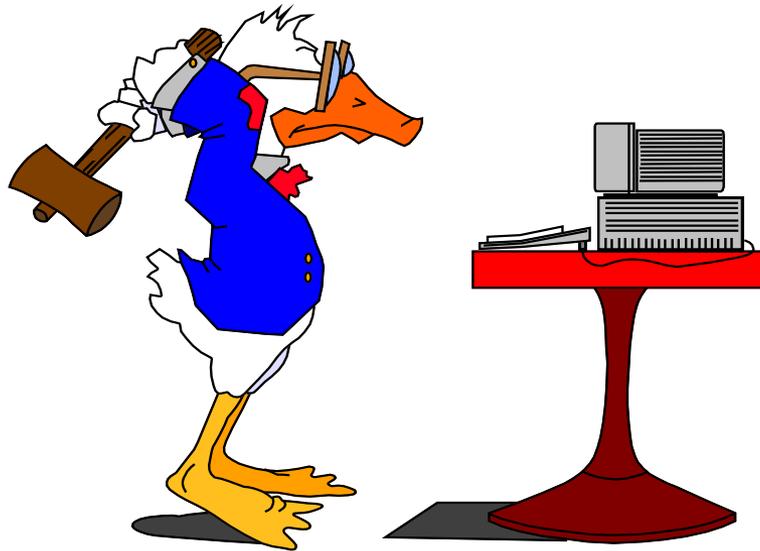
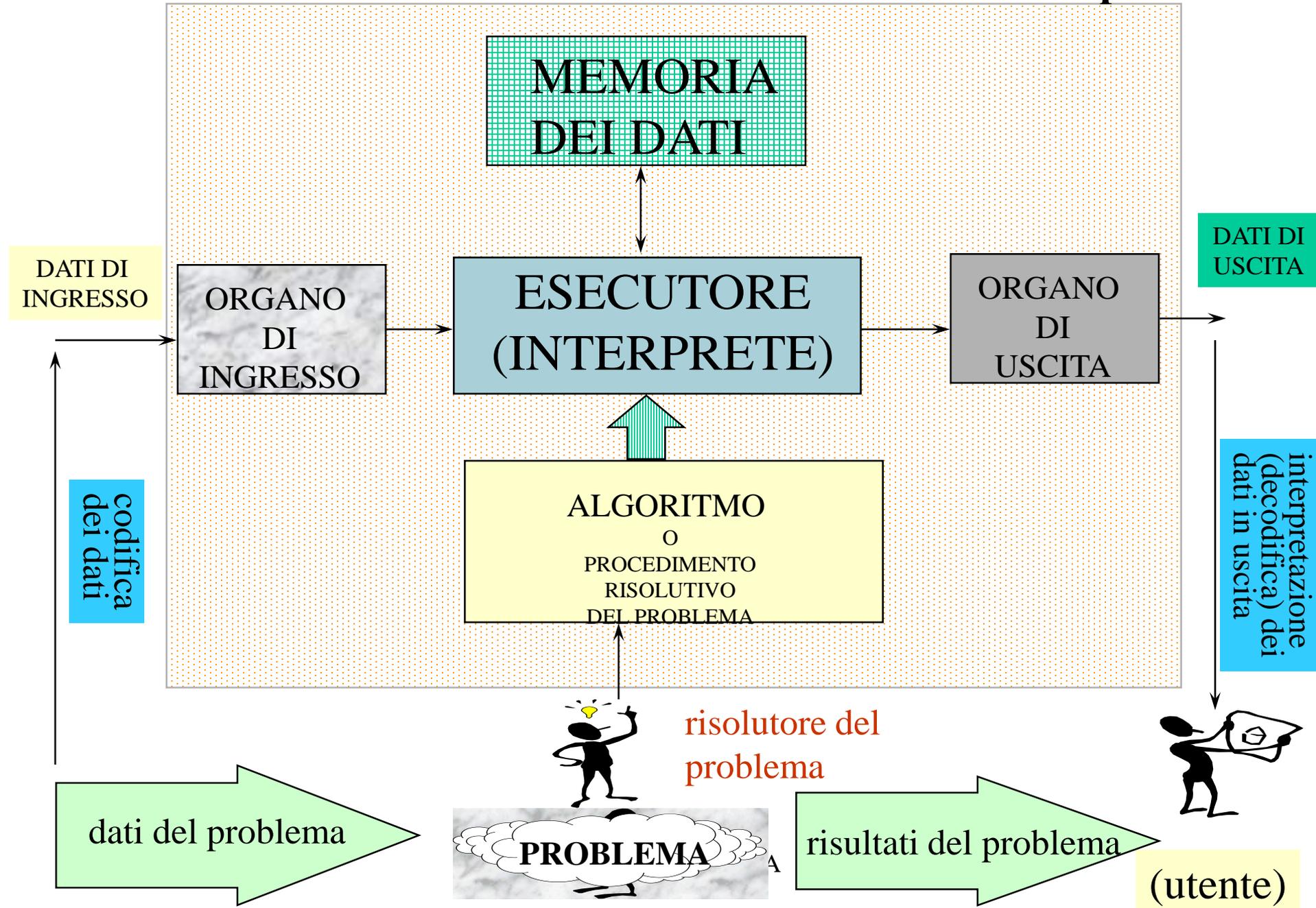


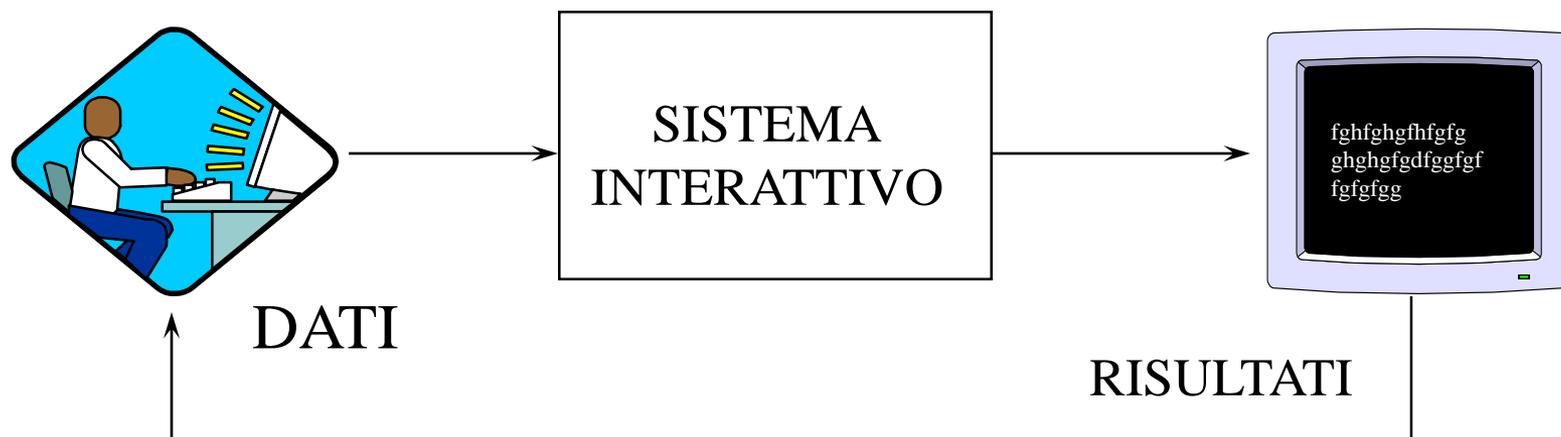
LA PROGRAMMAZIONE



Il sistema di calcolo nell'ambiente di risoluzione di un problema



Uso interattivo e non interattivo di un sistema di calcolo



modo non interattivo: l'utente introduce i dati di ingresso tutti in una volta e preleva i risultati tutti in una volta.

modo interattivo: l'utente introduce una parte dei dati, attende un primo risultato, in base al quale decide quale nuovo ingresso fornire e così via fino al termine (eventuale) del “*colloquio*”: esiste cioè una *interazione* fra l'utente e la macchina.

Descrivere algoritmi in Linguaggi Naturale comporta un vantaggio poiché esso è patrimonio comune ad un vasto numero di persone ma ha lo svantaggio di essere in generale:

- ***non preciso*** (e quindi ambiguo);
- ***complesso*** (difficilmente formalizzabile);
- ***inadeguato*** (a descrivere molti problemi);

Si deve però rilevare che durante le fasi preliminari di *analisi* di grossi problemi il *LN* è tuttora usatissimo.

Un **programma** è l'espressione di una soluzione (fra le tanti possibili) di una determinata classe di problemi in un determinato *linguaggio di programmazione*.

Un **linguaggio di programmazione** non può, naturalmente, risentire delle ambiguità insite nel LN proprio perché deve consentire di esprimere univocamente dei *metodi computazionali*.

Un **PROGRAMMA = ALGORITMO = METODO COMPUTAZIONALE** è un insieme di istruzioni espresse secondo *precise regole sintattiche* cui sono associate precise azioni di un *preciso esecutore*.

Conveniamo di chiamare **ambiente** un insieme di *variabili* con i valori loro associati.

Una **variabile** è un nome simbolico associato a numeri o a stringhe di caratteri, nel più basso livello implementativo (*linguaggio macchina*) essa rappresenta una *locazione di memoria*.

Si consideri l'*espressione*:

$$(3X - Y) * (X + 2)$$

dove

X e *Y* sono *variabili*

3 e 2 sono *costanti*

Il **valutatore** dovrà essere in grado di riconoscere e di accedere a tutte le componenti di un'espressione come le costanti, le variabili, e i simboli di operazioni, rispettando una ben precisa **sintassi**.

L'operazione di assegnamento

$\langle nome-var \rangle \longleftarrow \langle E \rangle$

$X \longleftarrow X + 1$

assegna ad X il valore ..

dove $\langle nome-var \rangle$ sta per un generico nome legale di variabile ed $\langle E \rangle$ rappresenta una generica espressione.

- la *lettura* di una variabile, cioè il processo di valutazione di una variabile è “**non distruttiva**”;
- la *scrittura* di una variabile, cioè l'assegnamento di un valore alla variabile, è invece “**distruttivo**”.

Problema: date due variabili A e B scambiare i contenuti.

una soluzione immediata ma errata è formata dalla coppia di istruzioni:

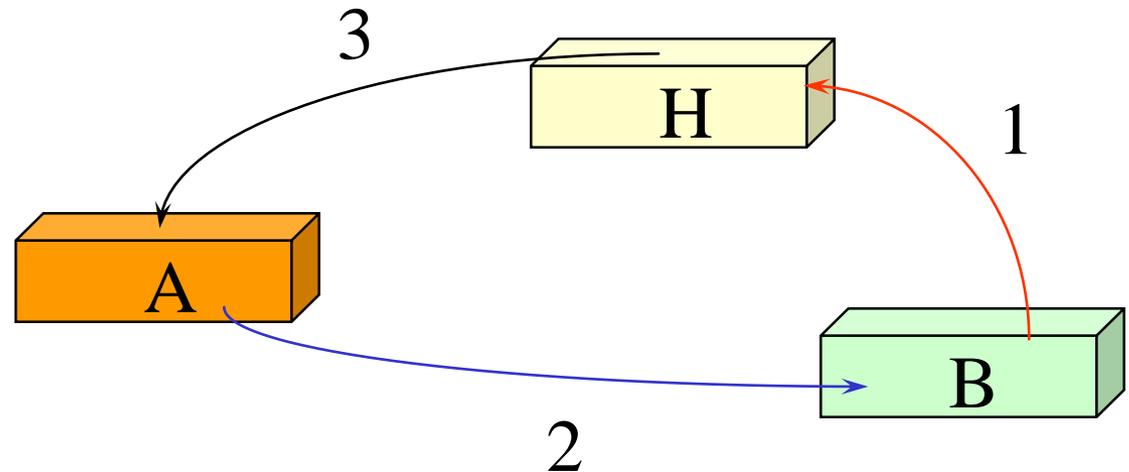
$$B \leftarrow A$$
$$A \leftarrow B$$

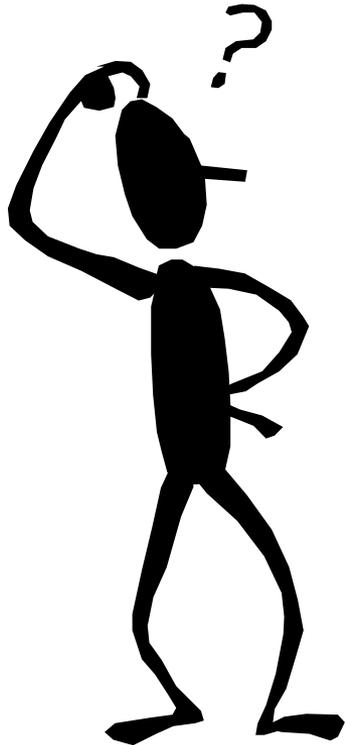
si perde il contenuto di B. Dobbiamo perciò salvare B in una variabile ausiliaria (H), si avrà:

1) $H \leftarrow B$

2) $B \leftarrow A$

3) $A \leftarrow H$





che valore ha una variabile prima del primo assegnamento ?

Esistono sostanzialmente tre tipi di risposta:

- *valore qualsiasi;*
- *un valore ben preciso, ad esempio zero;*
- *un valore speciale es. “?” che serve a segnalare una situazione di errore per “variabile non inizializzata”.*

Esercizi e Problemi

1. Si consideri la seguente sequenza di assegnamenti:

.....
4) $A \leftarrow B + 1$
5) $C \leftarrow B - 1$
6) $A \leftarrow D/2$
.....

Quale dei tre deve considerarsi errato o, quanto meno, “strano”?

2. Descrivere l’ambiente dopo l’esecuzione della seguente sequenza di assegnamenti?

1) $ZERO \leftarrow 0$
2) $UNO \leftarrow ZERO + UNO$
3) $DUE \leftarrow UNO + UNO$

3. Tre variabili A, B, C contengono rispettivamente i numeri 3, 1 e 2. Scambiare i loro contenuti in modo che contengano rispettivamente i numeri 1, 2 e 3.

4. Sia N una variabile contenente un intero $n < 10000$ e I una variabile contenente un intero compreso fra 1 e 4 inclusi. Scrivere un'espressione che esprima la I-esima cifra (da destra) di N (es. se $N = 4725$ e $I = 3$ l'espressione, valutata, fornisce 7).



$$\frac{N}{10^{I-1}} \bmod 10$$

Le istruzioni di Ingresso/Uscita

leggi $\langle nome-var \rangle$ ha l'effetto di assegnare alla variabile indicata il valore letto tramite l'organo d'ingresso.

Es. **scanf**("%d", &X)

Inserisce nella variabile di indirizzo X un valore intero

scrivi $\langle expr_1 \rangle, \langle expr_2 \rangle, \dots \langle expr_n \rangle$

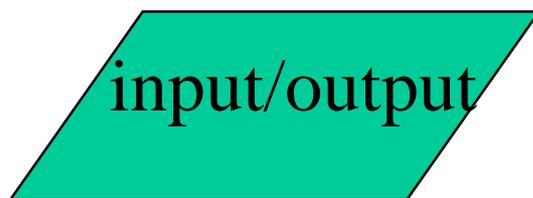
ha l'effetto di scrivere all'esterno, mediante l'organo di uscita, le valutazioni delle espressioni corrispondenti.

Es. **printf**("Il valore calcolato è %d ", X)

stampa il valore intero di X nello standard di output

Il linguaggio dei diagrammi a blocchi o flowcharts: un primo strumento per descrivere il controllo

Uno dei più diffusi formalismi per la descrizione del controllo degli algoritmi, si avvale di una dislocazione bidimensionale di blocchi contenenti le istruzioni e connessi mediante frecce di riferimento.



Una condizione è un confronto fra due expr:

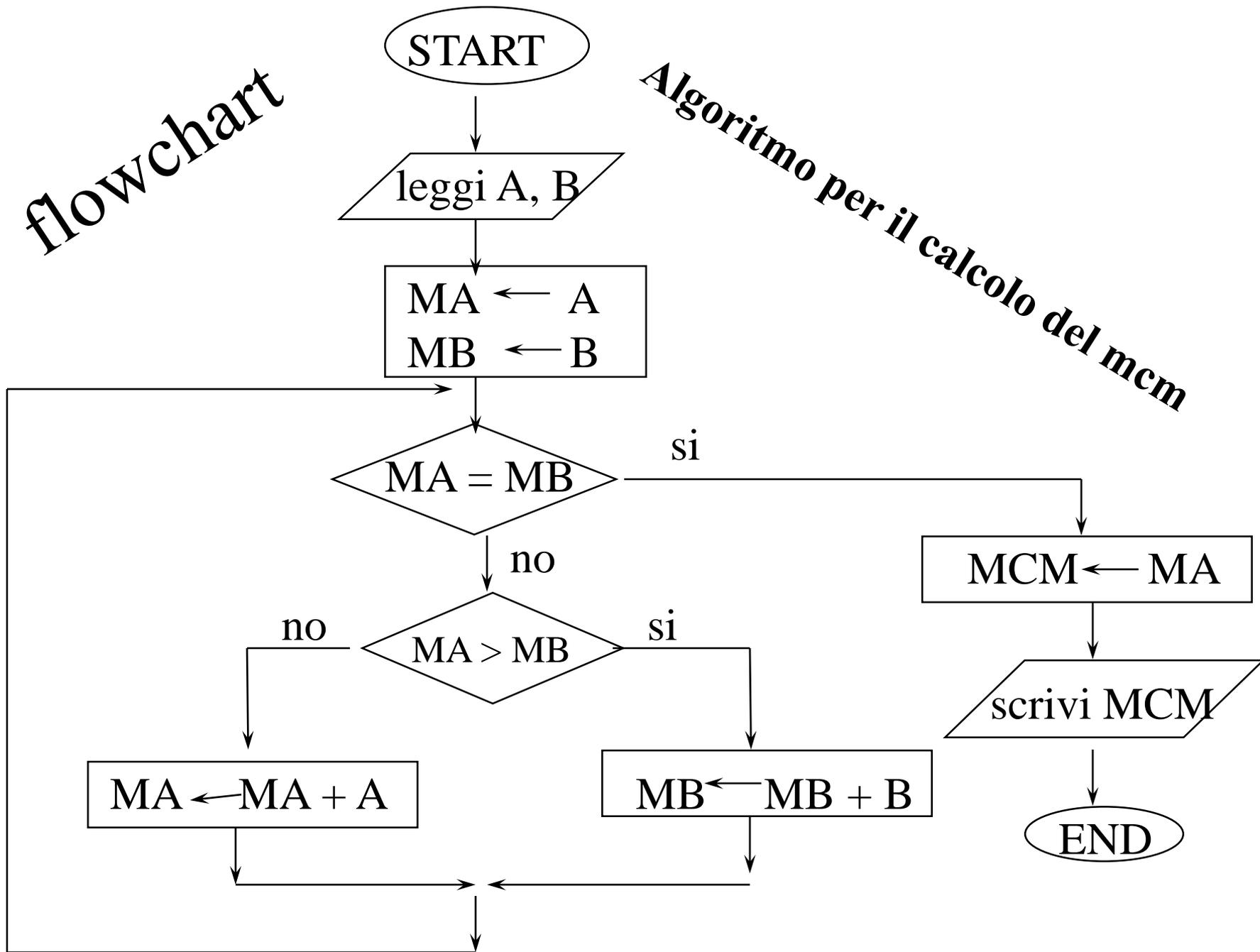
$$\langle \text{expr}_1 \rangle \langle \text{op_rel} \rangle \langle \text{expr}_2 \rangle$$

$\langle \text{op_rel} \rangle$ è uno dei 6 operatori relazionali:

$==, !=, >=, <=, >, <$

flowchart

Algoritmo per il calcolo del mcm

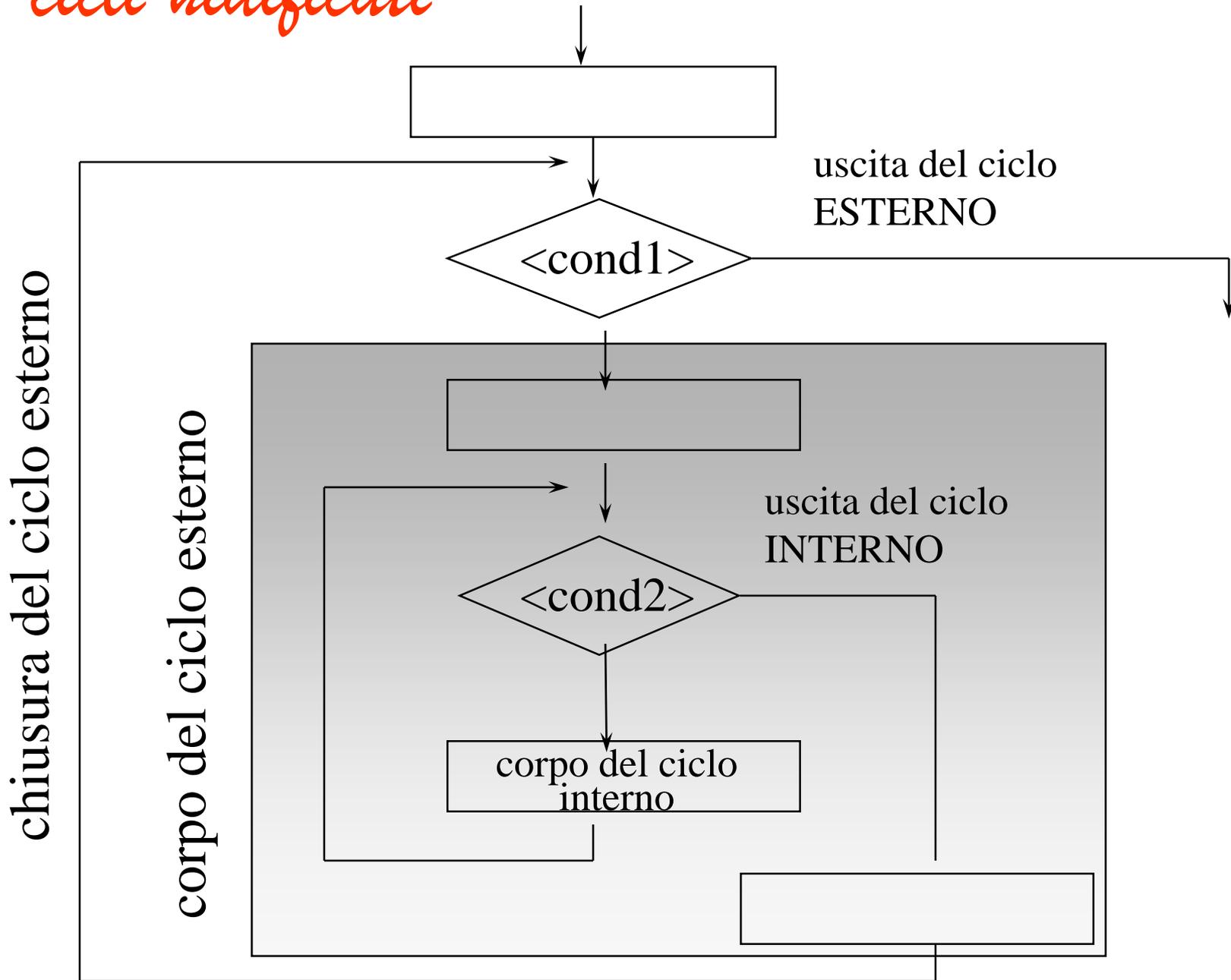


Affinché i cicli di qualsiasi tipo diano luogo a computazioni che *terminano correttamente* si devono rispettare le seguenti condizioni:

- assicurarsi dei valori iniziali delle variabili dalla cui modifica dipende il termine del ciclo;
- deve esserci almeno un blocco, nel ciclo, in cui vengono modificate le variabili da cui dipende l'uscita dal ciclo;
- deve esserci almeno un blocco di controllo dal cui esito dipenda l'uscita dal ciclo.

Se tali condizioni non sono osservate vi può essere la possibilità di *loop infinito* (generalmente non desiderato).

cicli nidificati



I COSTRUTTI DI CONTROLLO

Costrutto sequenza

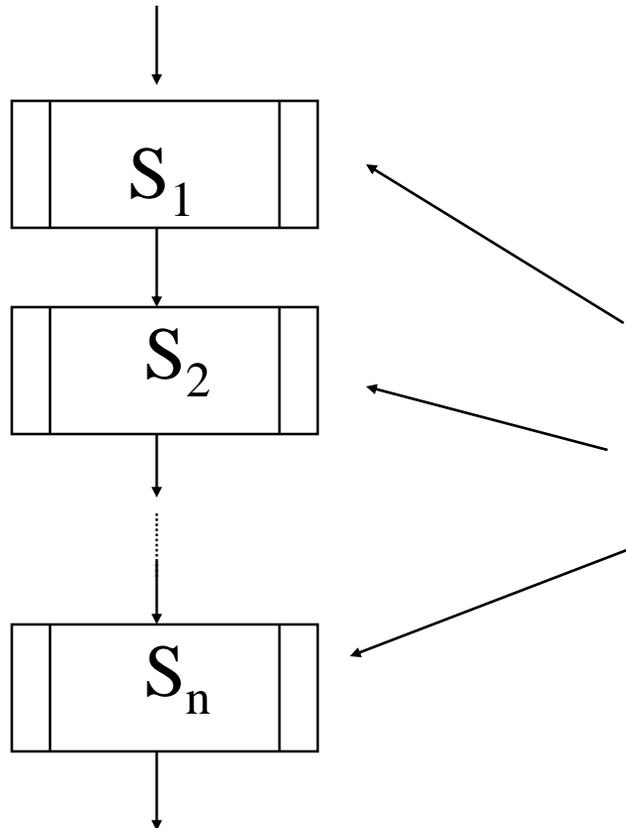
Sintassi:

$\langle S_1 \rangle$

$\langle S_2 \rangle$

$\langle S_n \rangle$

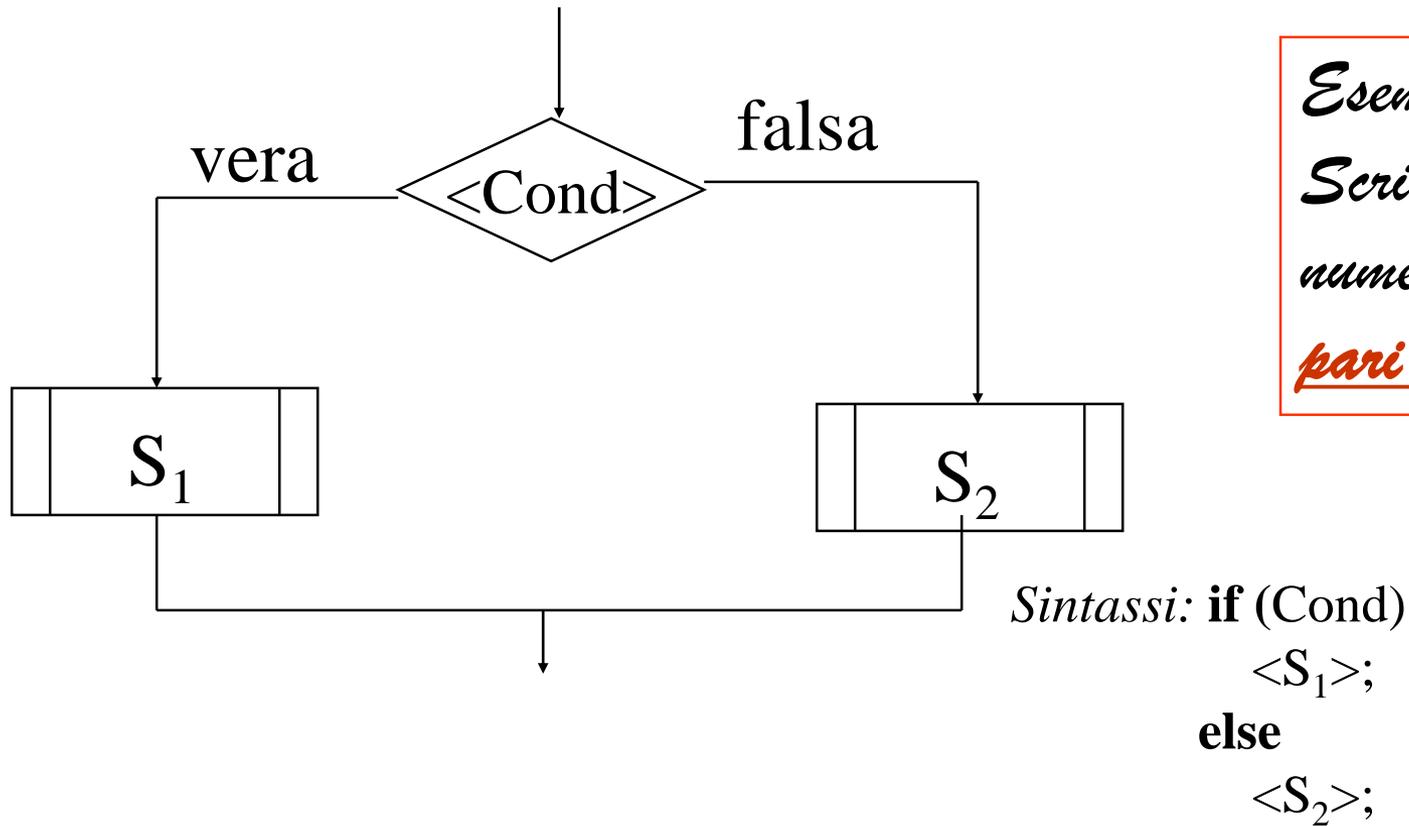
⋮



Blocchi strutturati
semplici o generati
da costrutti di
controllo strutturati
(**composti**)

Si impone di eseguire prima S_1 , poi S_2 fino ad S_n . Se uno qualsiasi dei blocchi va in loop infinito tutto il blocco va in loop infinito.

Costrutto **if-else** (*se-allora-altrimenti*)



Esempio:
Scrivere se un
numero dato è
pari o dispari.

Questo schema valuta inizialmente una condizione <Cond>; se questa è vera, viene eseguito un blocco strutturato <S₁> altrimenti viene eseguito un blocco strutturato <S₂>.

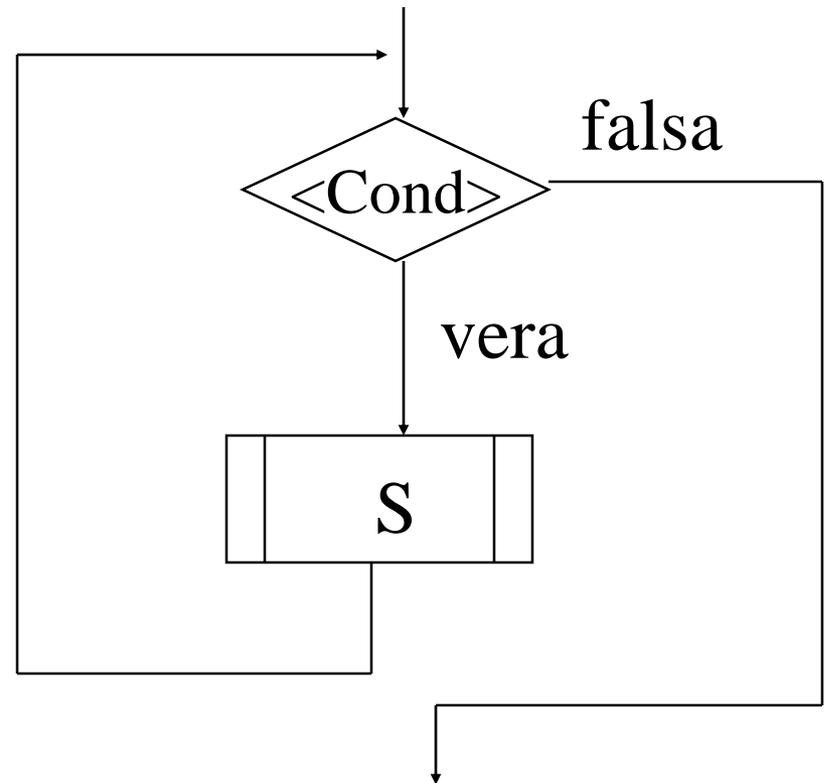
Costrutto **while** (finché la condizione è vera ripeti)

Sintassi:

while (Cond)

<S>;

Questo schema prevede una condizione <Cond> (detta guardia del ciclo) da valutare inizialmente. Se questa è falsa si esce dal ciclo, altrimenti si esegue ciclicamente un blocco strutturato <S> detto corpo del while.



*Esempio: calcolo del **m.c.m.** fra due numeri naturali A e B*

E' importante citare un importante risultato dovuto a due ricercatori italiani, e da cui il teorema seguente è solo una conseguenza.

Teorema (di Böhm-Jacopini)

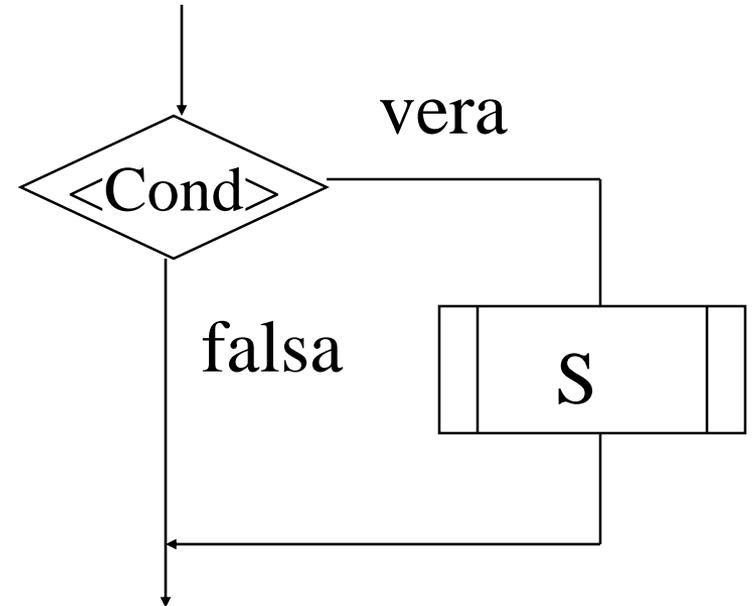
I costrutti di controllo *sequenza*, *if-else* e *while* sono sufficienti ad esprimere un diagramma equivalente a qualsiasi altro, dal punto di vista dei processi generati, per qualsiasi valori dei dati in ingresso.

Costrutto **if** (*se-allora*)

Sintassi:

1) **if** (Cond) <S>;

2) **if** (Cond)
 <S>;

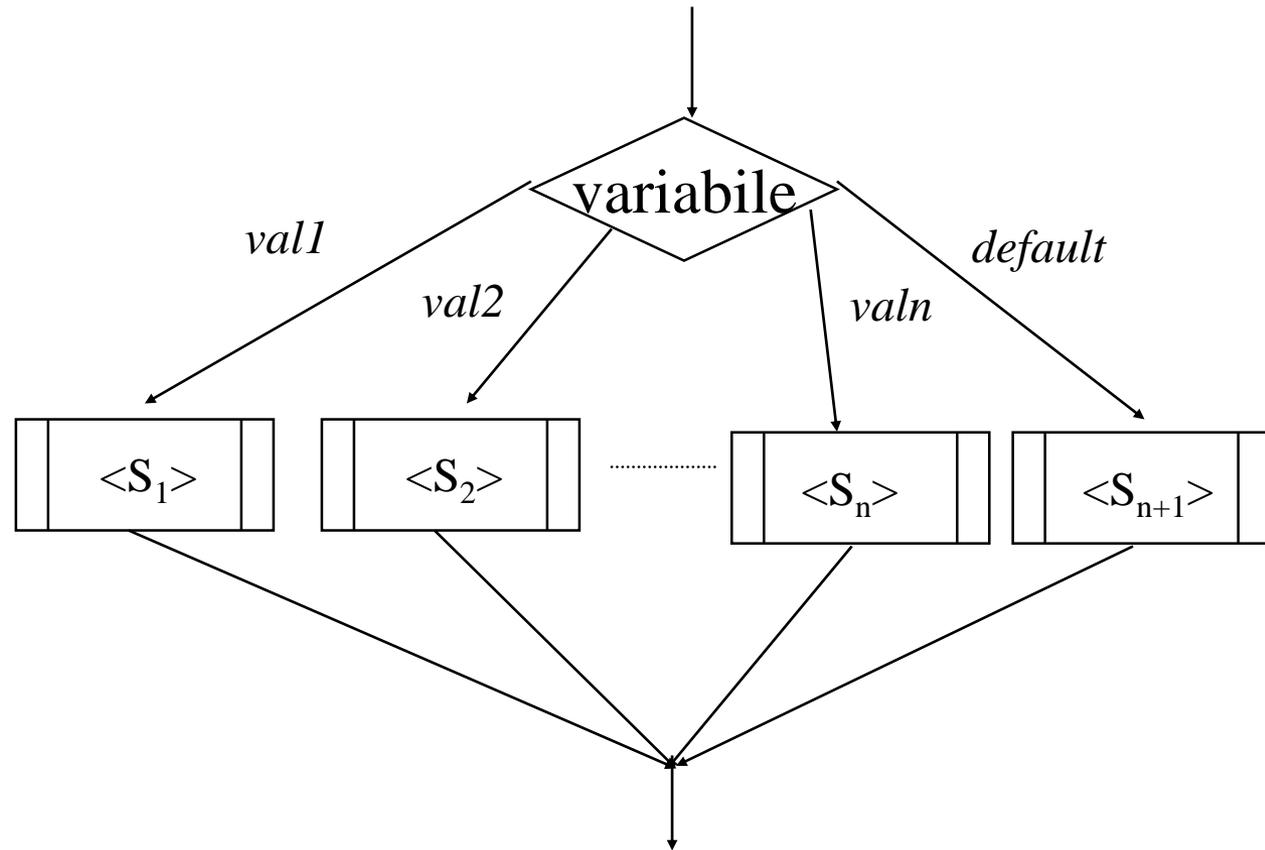


Esempio: **if** (A > MASSIMO) MASSIMO=A;

Costrutto switch (selezione a molte vie)

Sintassi:

```
switch(variabile){  
  case: val1:  
    istruzioni-1;  
    break;  
  case: val2:  
    istruzioni-2;  
    break;  
  ..  
  case: valn:  
    istruzioni-n;  
    break;  
  default:  
    istruzioni;  
}
```



Nel costrutto *case*, l'esecutore non esegue un confronto sequenziale di tutti i casi ma è bensì in grado di individuare subito il blocco da eseguire. Inoltre la parte *default* può mancare.

Costrutto **DO ... While**(ripeti finché è vero)

Sintassi:

Do

{

$\langle S_1 \rangle;$

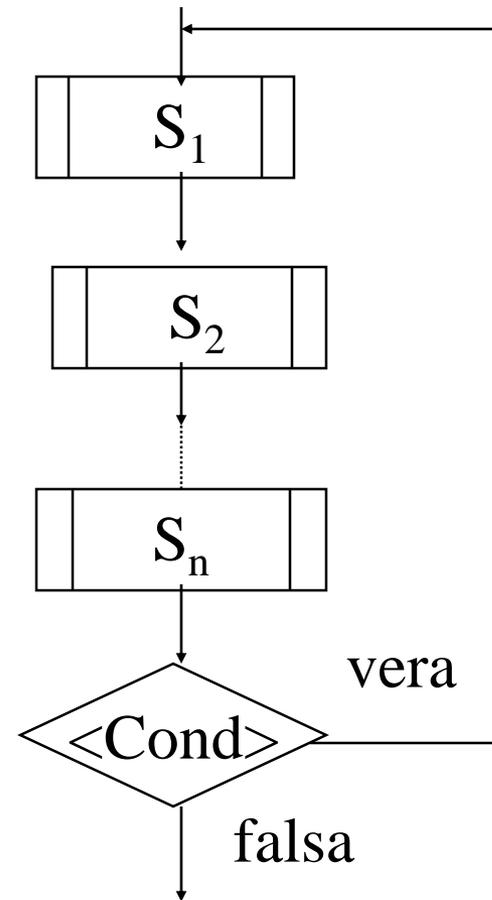
$\langle S_2 \rangle;$

⋮

$\langle S_n \rangle;$

}

While (Cond);

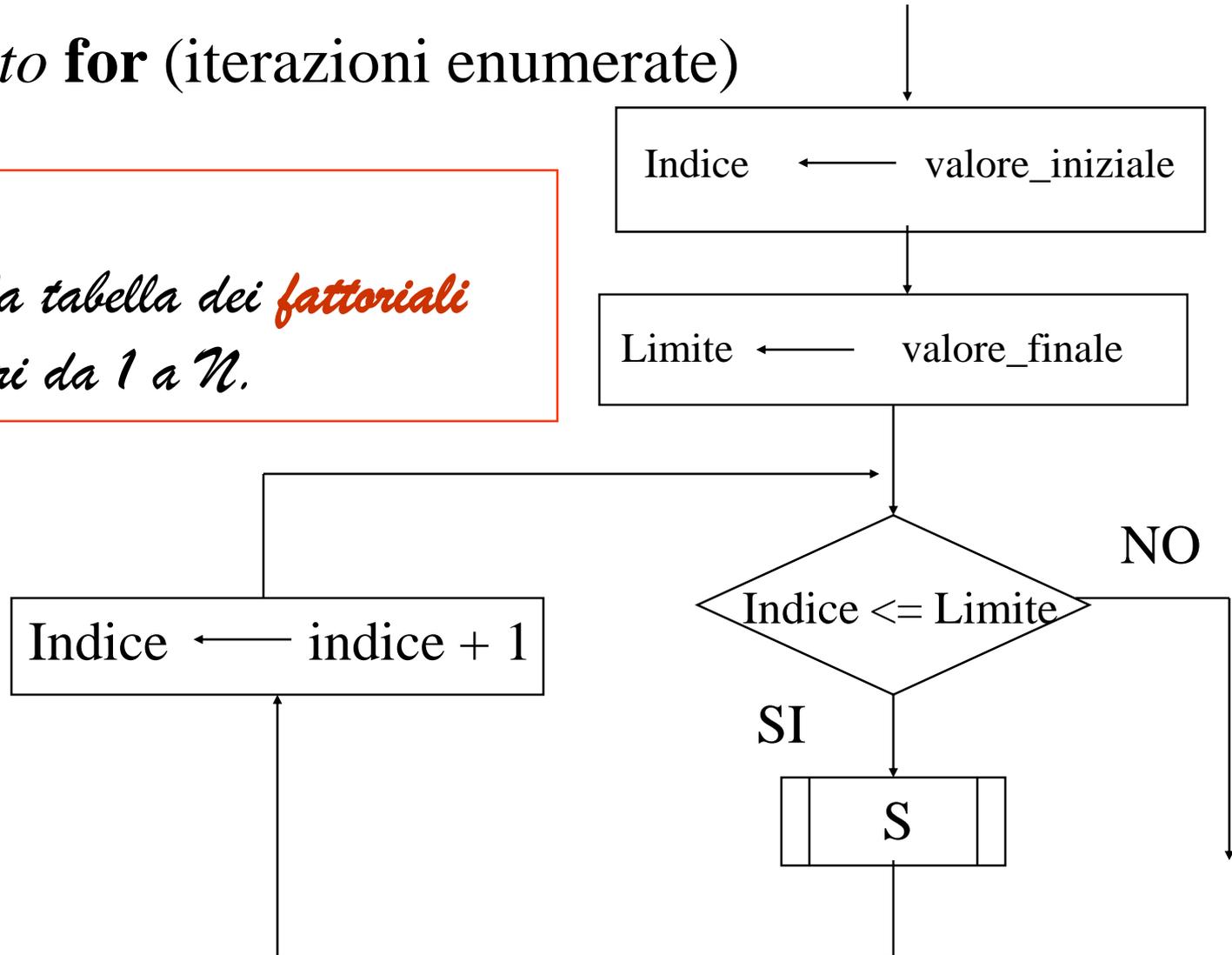


*Esempio: Dato un elenco di persone, con l'indicazione di ciascuna di esse del nome e dell'età, si devono **contare** i maggiorenni.*

Costrutto **for** (iterazioni enumerate)

Esempio:

*Scrivere la tabella dei **fattoriali**
dei numeri da 1 a N .*



Sintassi:

```
for (indice = valore_iniziale; indice < valore_finale; incremento indice)  
  <S>;
```

Utilizzare Algotool per scrivere i seguenti FlowChart

1. Dati due numeri interi A e B scrivere prima il minore e poi il maggiore.
2. Dati due numeri interi A e B stampare la loro somma, il loro prodotto, la differenza, il quoziente e resto della divisione intera.
3. Dato un numero N dire se è pari, dispari, negativo o positivo.
4. Dati tre numeri A, B, C scambiarli in maniera tale che siano ordinati dal più piccolo al più grande.
5. Calcolare la somma dei primi N numeri interi positivi.
6. Inserire da tastiera N numeri interi, contare quanti ne sono positivi e quanti negativi.
7. Inserire da tastiera N numeri interi, contare quanti ne sono PARI e DISPARI.
8. Calcolare la media dei numeri interi compresi tra -7 e $+15$.
9. Dato un numero N visualizzare i suoi divisori.
10. Calcolare il prodotto tra A e B con somme successive.