

# Algoritmi

---

# Algoritmi

---

- Risolvere un problema significa individuare un procedimento che permetta di arrivare al risultato partendo dai dati
- Il procedimento (chiamato *algoritmo*) è composto da passi elementari

# Esempi di algoritmi

---

## ***Ricetta per cucinare gli spaghetti***

- Metti l'acqua nella pentola
- Fai bollire l'acqua
- Metti la pasta nell'acqua
- Aggiungi un po' di sale
- Attendi 6 minuti
- Scola la pasta

# Esempi di algoritmi

---

## ***Verifica se un numero è pari o dispari***

- Prendi il numero
- Calcola il resto della divisione intera del numero per 2
- Se il resto è zero
  - Allora il numero è pari
  - Altrimenti il numero è dispari

# Esempi di algoritmi

---

## Verifica se un numero e' primo:

1. Prendi il numero  $n$
2. Poni un contatore "i" a 2
3. Calcola il resto della divisione intera  $n/i$
4. Se il resto è zero
  - Allora il numero NON e' primo
  - Altrimenti, se il contatore e' minore di .... allora incrementa il contatore  $i$  e vai al passo 3

- *Dubbio: fino a quanto dobbiamo incrementare il contatore ??*

# Proprietà degli algoritmi

---

- L'algoritmo è caratterizzato da:
  - **finitezza**: composto da un numero finito di passi elementari; le operazioni sono eseguite un numero finito di volte
  - **non ambiguità**: i risultati non variano in funzione della macchina/persona che esegue l'algoritmo (deterministico)
  - **realizzabilità**: deve essere eseguibile con le risorse a disposizione

# Proprietà degli algoritmi

---

- *... ma gli esempi precedenti possiedono queste proprietà?*
- Problemi presenti:
  - Ambiguità
  - Ipotesi implicite sulle capacità dell'esecutore
  - Uso del linguaggio naturale

# Rappresentazione degli algoritmi

---

- E' necessario far riferimento a dei formalismi che:
  - non introducano ambiguità e siano universalmente riconosciuti ed interpretati allo stesso modo da un generico esecutore
  - permettano di rappresentare in modo efficace un algoritmo
  - Costituiscono un utile strumento per poi poter passare alla fase di codifica in un linguaggio di programmazione

# Rappresentazione degli algoritmi

---

Rappresentazione grafica

Diagrammi di Flusso / Flow Chart

Rappresentazione testuale

Notazione Lineare Strutturata / PseudoCodice

# Algoritmi: operazioni base

---

- Le operazioni primarie sono:
  - **Istruzioni di I/O** (Trasferimento di informazioni: lettura dati, scrittura risultati)
  - **Istruzioni di assegnamento**
  - **Valutazione espressioni** (esecuzione di calcoli)
  - **Strutture di controllo** (che modificano il flusso sequenziale di esecuzione delle operazioni)

# Flow chart: simboli fondamentali

---



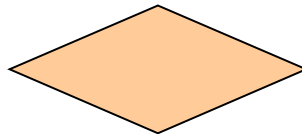
**ingresso/uscita**



**inizio/fine**



**Elaborazione**



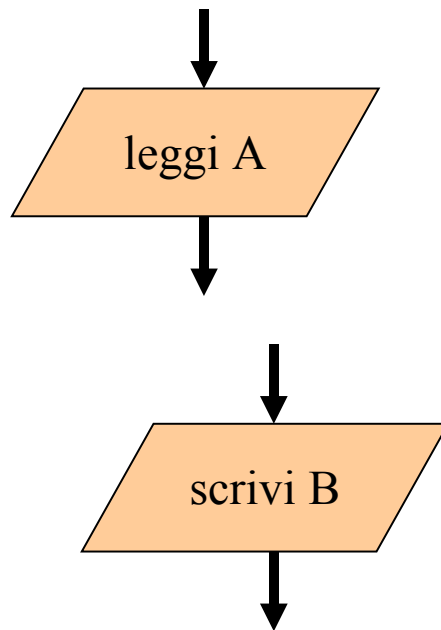
**Selezione**

*...ce ne sono anche altri con forme diverse di importanza secondaria, ma questo non e' un corso di disegno...*

# Istruzioni di I/O

- lettura di dati in input
- scrittura dei risultati in output

Flow chart



Pseudo-codice

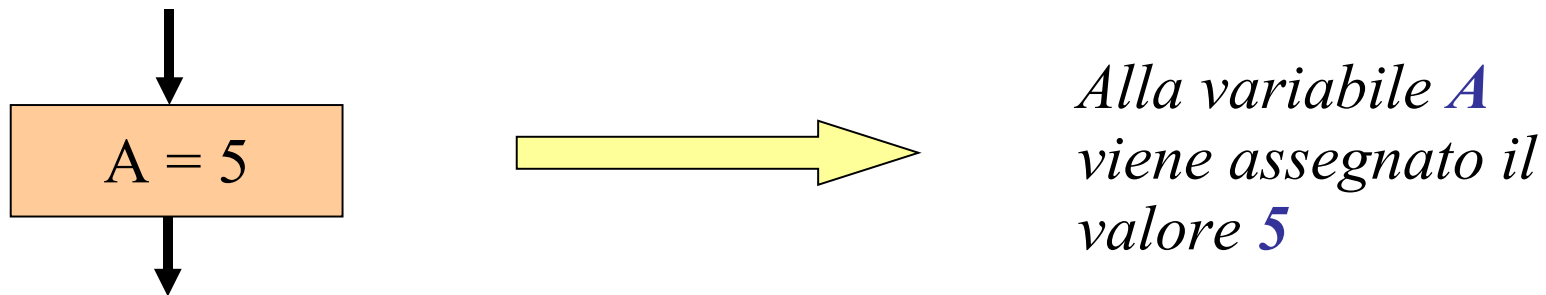
leggi A

scrivi B

# Istruzione di assegnamento

---

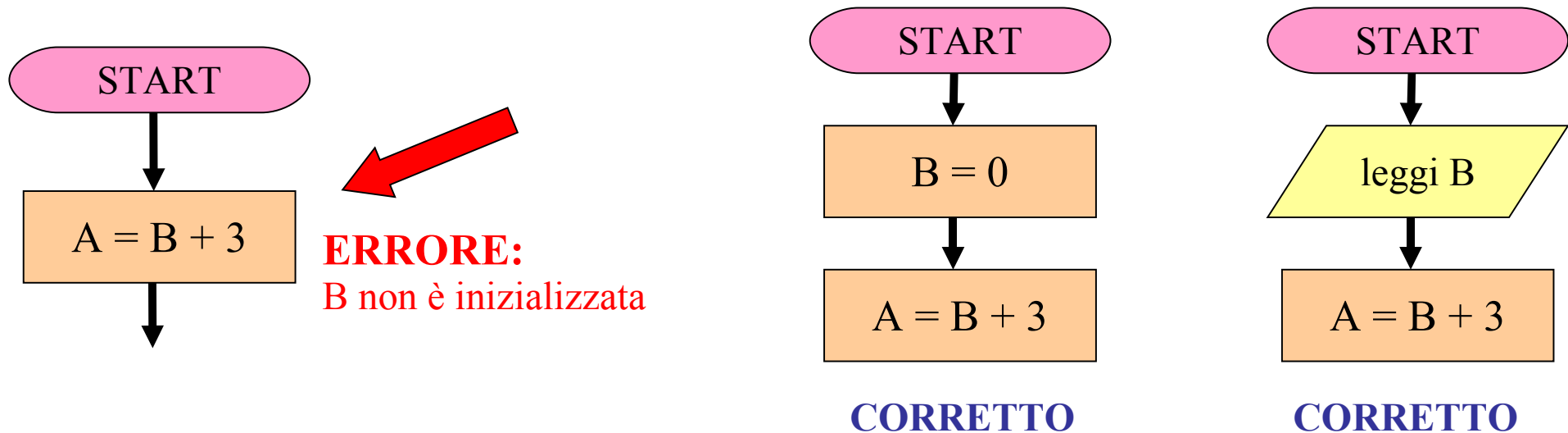
- Concetto di **variabile**
  - Identificata da un'etichetta / identificatore simbolico
  - Può essere comodo pensare alla variabile come ad un contenitore in cui possiamo memorizzare o reperire dei dati utilizzati durante il calcolo
  - L'istruzione di assegnamento permette di **assegnare** un valore ad una variabile



# Istruzione di assegnamento

## Inizializzazione

- All'inizio di un algoritmo una variabile non ha alcun valore
- Non ha quindi senso utilizzarla in una espressione (**è un errore!**)
- Essa deve essere inizializzata:
  - O esplicitamente mediante un assegnamento
  - Oppure mediante una operazione di lettura



# Istruzione di assegnamento

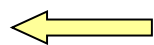
- E' stato usato il simbolo  $=$  per indicare l'istruzione di assegnamento (*sarà lo stesso usato nel linguaggio C*)
- Alcuni testi/autori indicano invece il simbolo  $\leftarrow$  (*per evitare confusione con l'operatore di uguaglianza*)

Dato il seguente frammento di codice:

...

$A = 5$

$A = A + 1$



Cosa fa quell'istruzione?

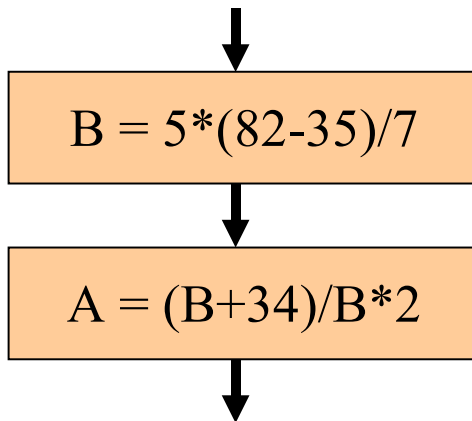
L'esecutore esegue i seguenti passi:

- prima valuta l'espressione a destra, cioè calcola il valore di  $A+1$  che vale 6
- dopo assegna tale valore alla variabile a sinistra, cioè ad  $A$
- alla fine dell'esecuzione di quella istruzione quindi,  $A$  vale 6

# Valutazione espressioni

- L'esecutore è in grado di valutare espressioni aritmetiche ed assegnarle alla variabile specificata nella parte sinistra.

Diagramma a blocchi



Pseudo-codice

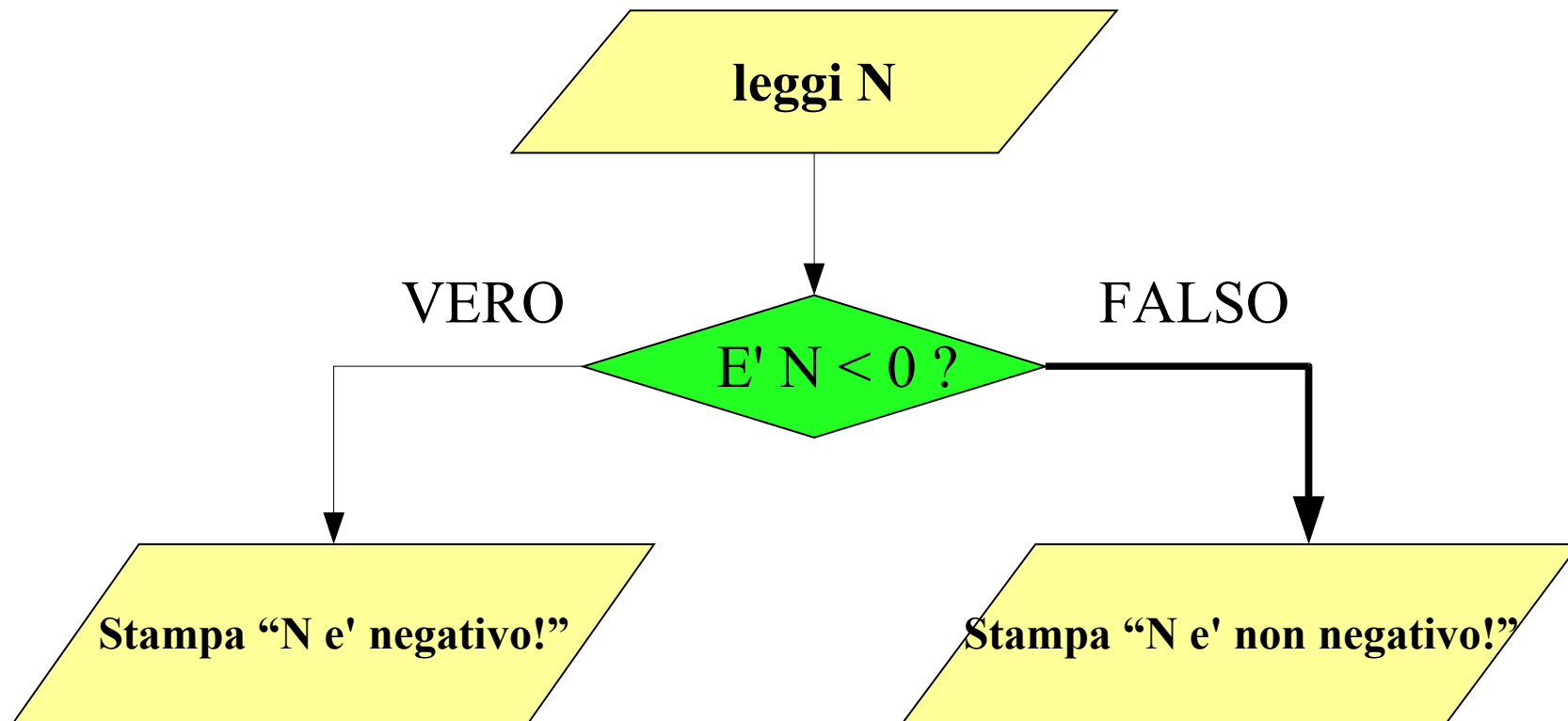
$B = 5*(82-35)/7$

$A = (B+34)/B*2$

# Struttura di Selezione

---

- Verifica di una condizione



# Diagrammi di Flusso: Esercizi

---

- Leggere base  $B$  e altezza  $H$  e stampare l'area
- Leggere due numeri  $X$  e  $Y$ , stampare il maggiore dei due
- Leggere due numeri  $X$  e  $Y$ , e scambiare loro valori (ossia alla fine  $X$  contiene quello che c'era in  $Y$  e viceversa...)
- Leggere un numero e stampare un messaggio a seconda che sia primo o meno.