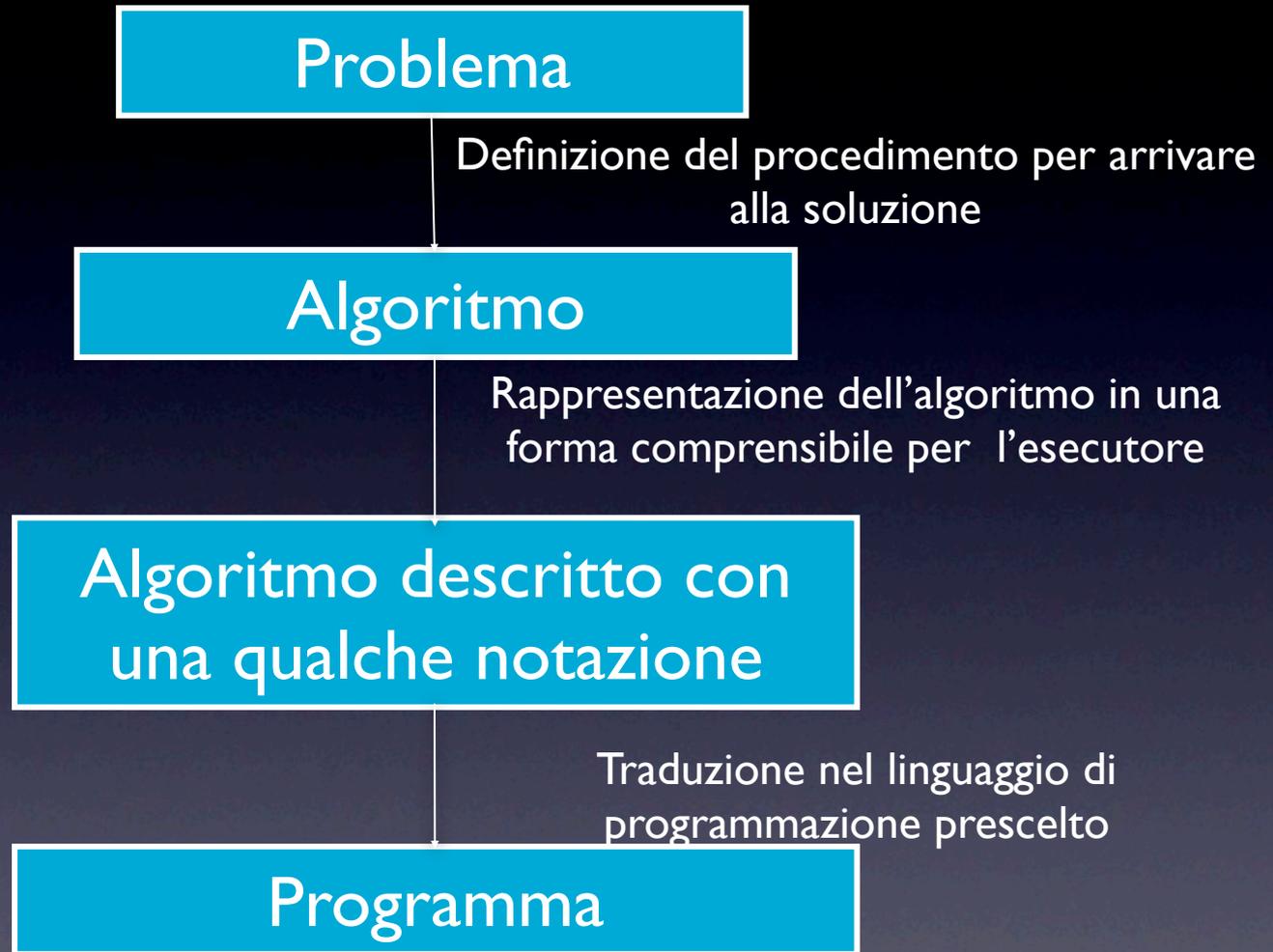


# Rappresentazione degli algoritmi

# Rappresentazione degli algoritmi



# Linguaggio naturale - Notazione lineare

- Nella rappresentazione di un algoritmo si possono riconoscere tre classi di istruzioni:
  - istruzioni di ingresso e uscita (dati e risultati del problema)
  - istruzioni di assegnamento ( $P \leftarrow A$  con il significato assegna al nome logico P il valore di A)
  - istruzioni di controllo: sono quelle istruzioni che modificano la sequenza dell'esecuzione
    - alterazione incondizionata
    - alterazione condizionata

# Esempio

Calcolare il prodotto di due numeri interi positivi  $A$  e  $B$  supponendo che l'esecutore conosca solo le operazioni di somma, sottrazione e confronto fra numeri.

## N.istruzione Istruzione

- 1 leggi  $A$  e  $B$
- 2  $P \leftarrow A$
- 3 se  $B$  è uguale 1 andare all'istruzione 7
- 4  $P \leftarrow P + A$
- 5  $B \leftarrow B - 1$
- 6 andare alla 3
- 7 scrivi  $P$
- 8 fine

# Tabella di Traccia

L'algoritmo può essere testato utilizzando una Tabella di Traccia.

Esempio: l'algoritmo precedente eseguito per i valori di ingresso 7 e 4

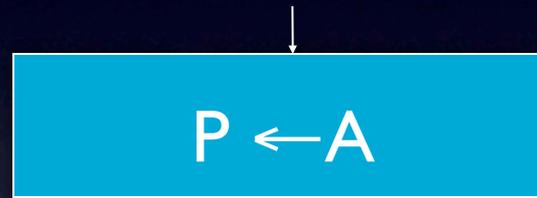
istruzione	A	B	P
inizio			
Leggi A e B	7	4	
$P \leftarrow A$			7
B è uguale a 1 ?			
$P \leftarrow P + A$			14
$B \leftarrow B - 1$		3	
B è uguale a 1 ?			
$P \leftarrow P + A$			21
$B \leftarrow B - 1$		2	
B è uguale a 1 ?			
$P \leftarrow P + A$			28
$B \leftarrow B - 1$		1	
B è uguale a 1 ?			
Scrivi P			

# Diagrammi a blocchi

Uno dei primi e più diffusi formalismi per la descrizione degli algoritmi, che si avvale di una dislocazione di blocchi contenenti le istruzioni e connessi mediante frecce, è detto Diagrammi a Blocchi.

Le forme dei blocchi sono:

**Rettangolo:** indica una operazione come ad esempio un assegnamento



**Rombo:** indica una diramazione, cioè la verifica di una condizione e la contrassegnazione degli archi uscenti in conformità con il risultato della verifica



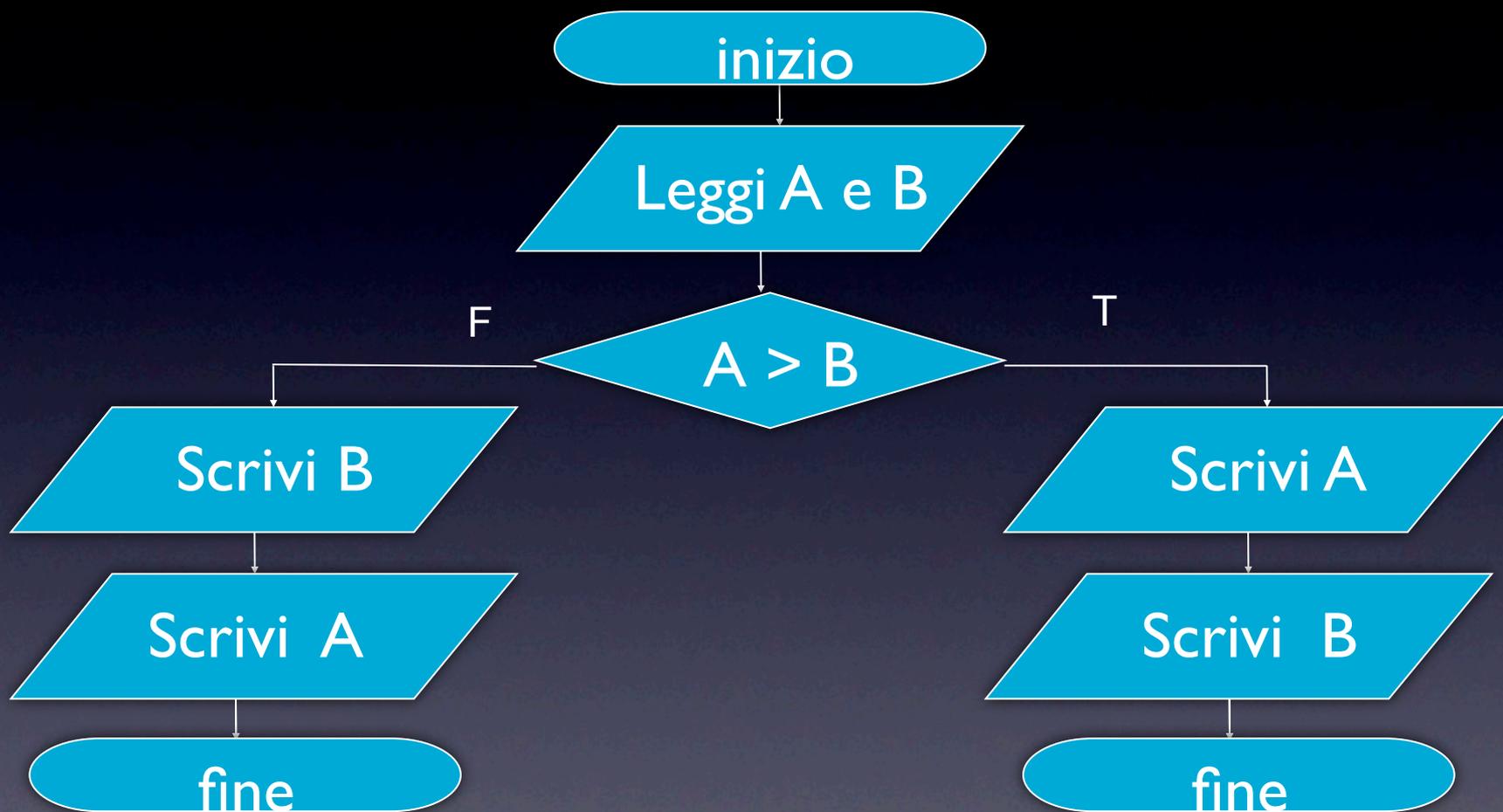
**Parallelogramma:** indica una operazione di ingresso o uscita



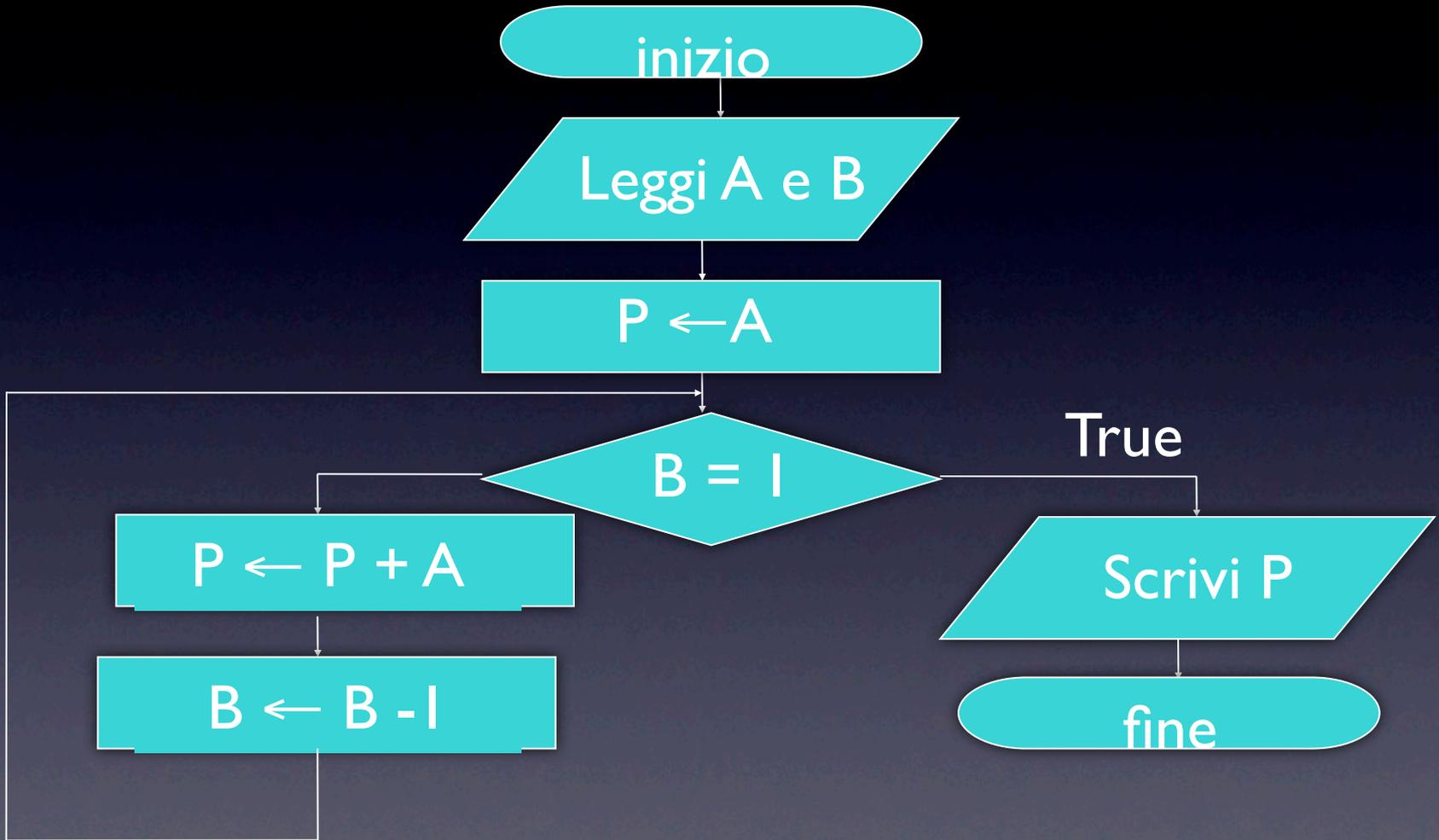
**Ellissi:** indica l'inizio o la fine dell'algoritmo



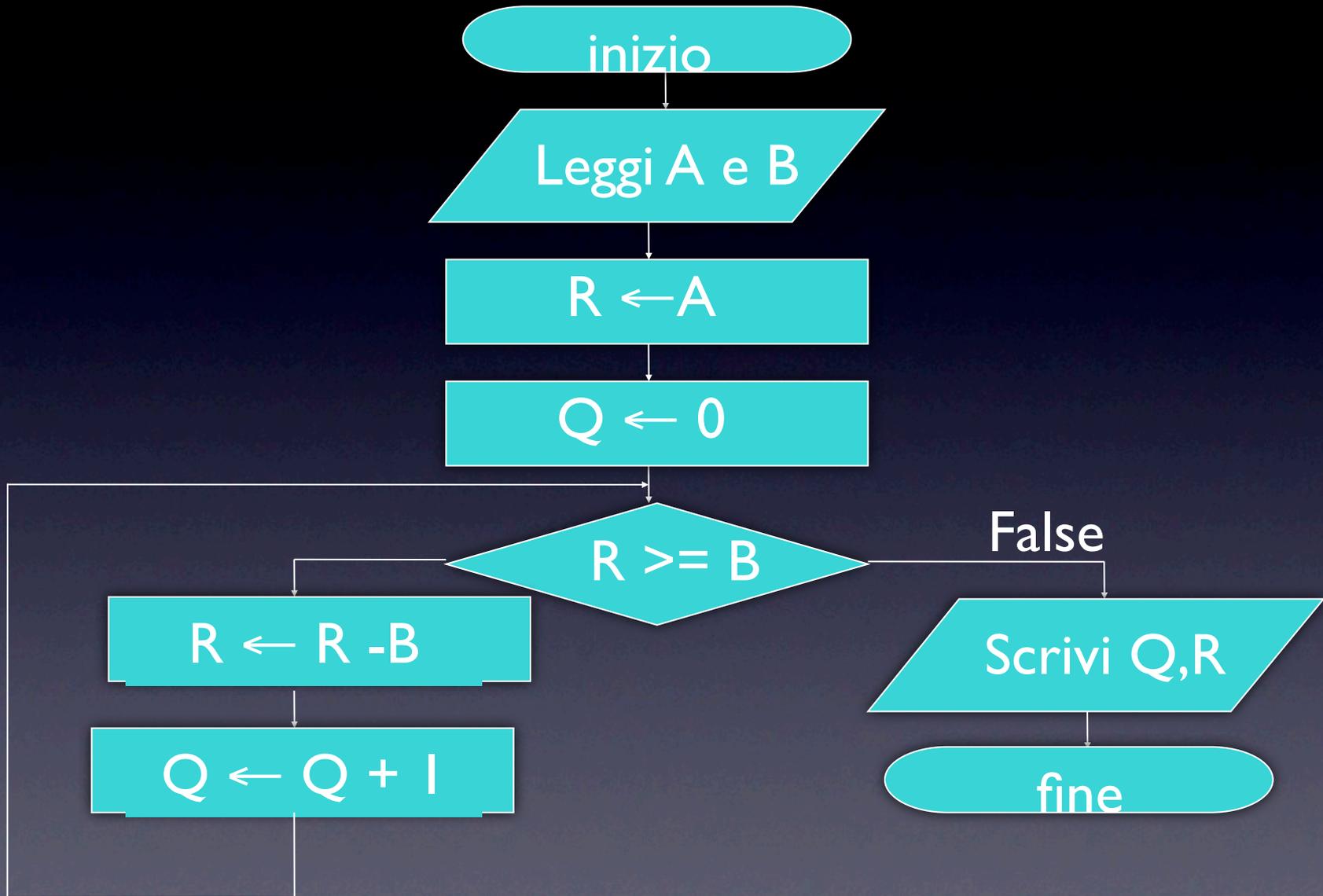
Dati due numeri interi, stampare prima il più grande e poi il più piccolo



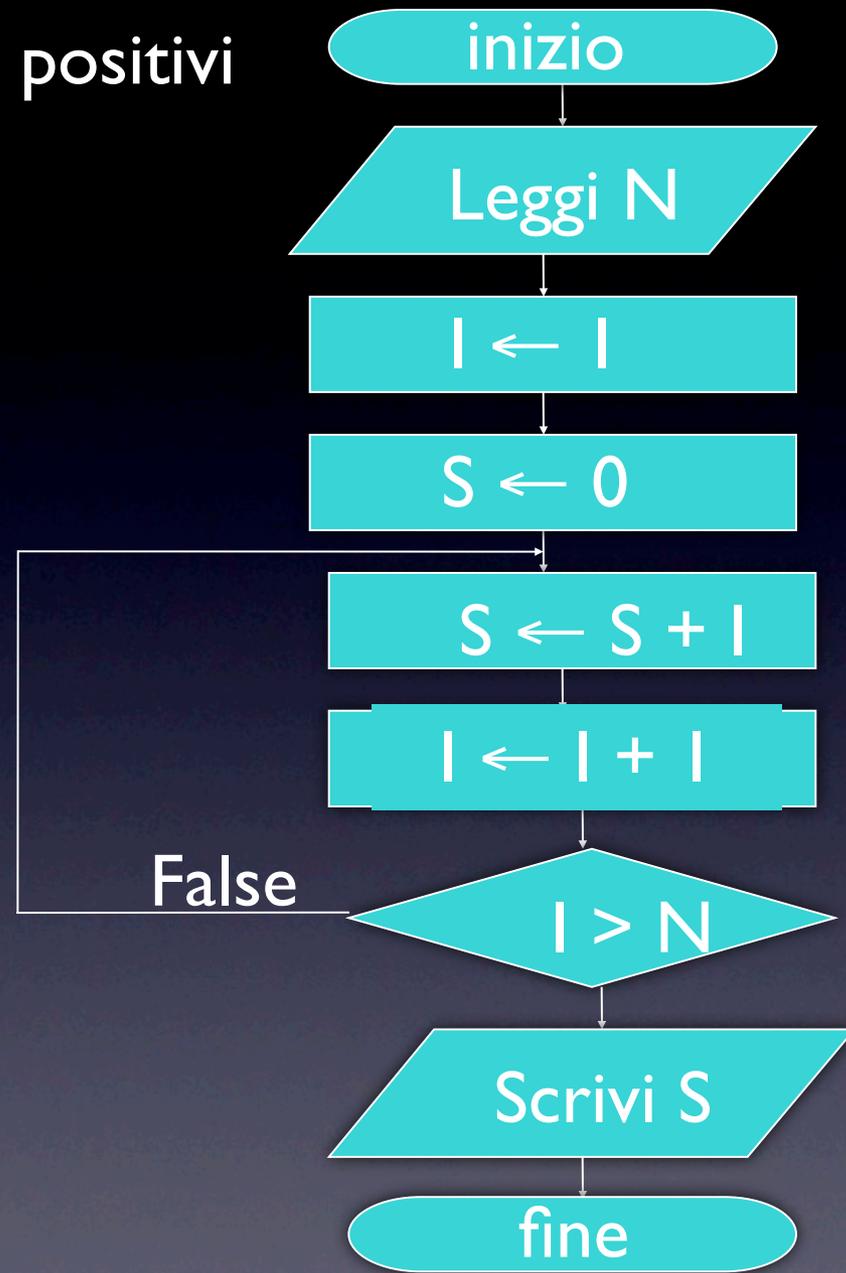
Dati due numeri interi positivi, effettuare il loro prodotto con il metodo delle addizioni successive.



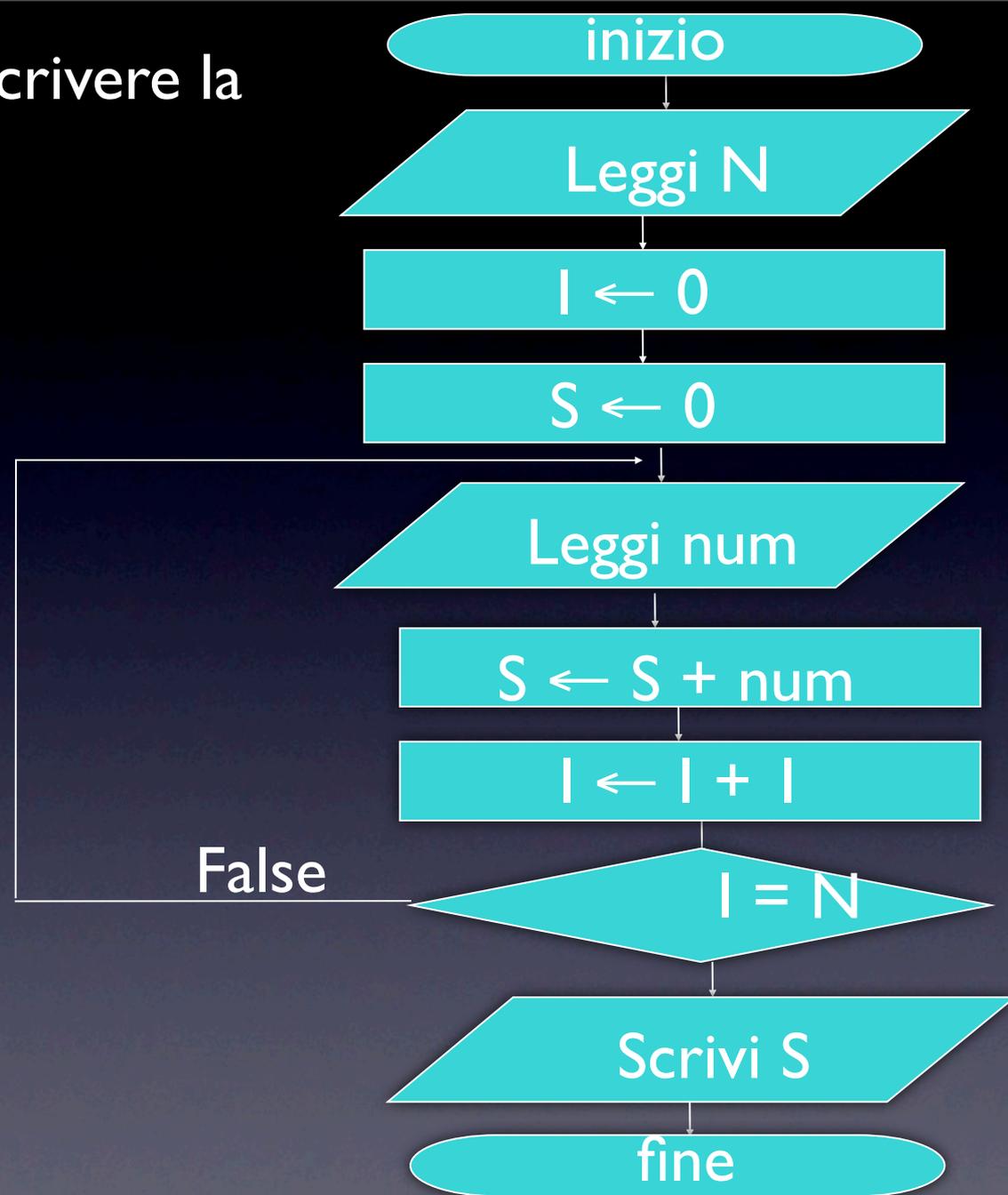
Dati due numeri interi positivi diversi da zero, calcolare quoziente e resto con il metodo delle sottrazioni successive



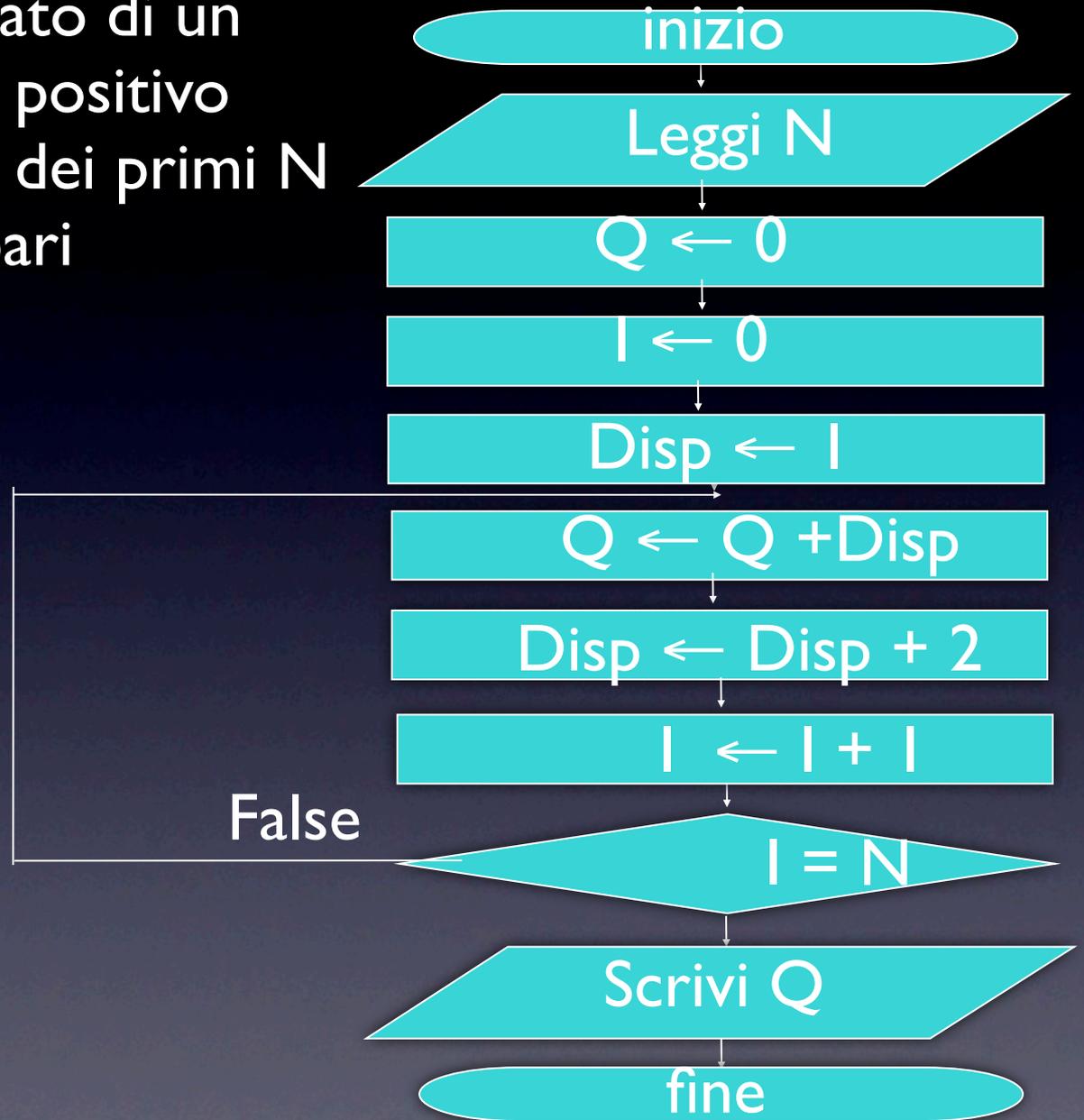
# Sommare i primi N interi positivi



Leggere N numeri e scrivere la loro somma.



Calcolare il quadrato di un numero  $N$  intero positivo utilizzando la somma dei primi  $N$  numeri dispari



# Limiti dei diagrammi a blocchi

- I DaB sono generalmente illeggibili, non si riesce a seguire l'algoritmo soprattutto quando superano le dimensioni di un semplice esercizio didattico. La lettura andrebbe fatta un po' dall'alto un po' dal basso senza un ordine preciso
- I DaB sono facilmente esposti ad errori logici, bastano pochi accavallamenti di cicli per perdere il filo del controllo. Ciò nasce dalle correzioni consentite da un indisciplinato uso delle frecce che causa un proliferare di errori logici che si accentua con la complessità del problema e l'inesperienza del risolutore
- Scarsa praticità dovuta alla natura grafica bidimensionale
- difficoltà di riconoscimento della struttura di controllo

*disciplina di composizione* che eviti cattive strutturazioni  
degli algoritmi

L'idea base è:

**un algoritmo deve essere letto dall'alto al basso secondo  
un ordine sequenziale di esecuzione**

Cio non significa che non possono esistere dei cicli o dei test, ma che questi siano strutturati in modo da poter essere considerati come un unico blocco operativo con un unico ingresso e una sola uscita.

Si possono distinguere

- blocchi semplici
- blocchi composti

Tutti con un unico punto di ingresso e un unico punto di uscita