

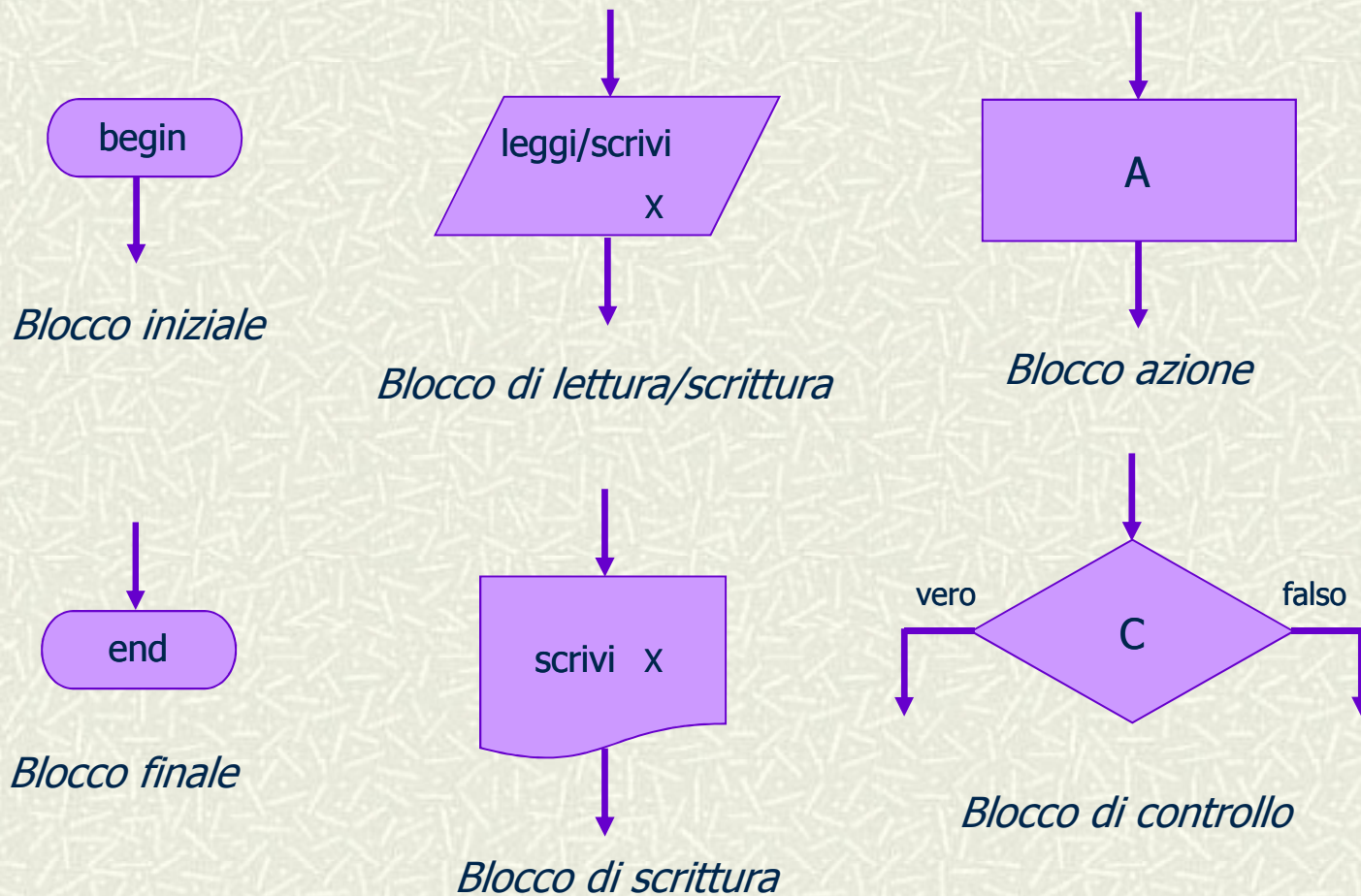
DIAGRAMMI A BLOCCHI

TEORIA ED ESERCIZI

I diagrammi a blocchi – 1

- # Il linguaggio dei **diagrammi a blocchi** è un possibile formalismo per la descrizione di algoritmi
- # Il diagramma a blocchi, o *flowchart*, è una rappresentazione grafica dell'algoritmo
- # Un diagramma a blocchi descrive il flusso delle operazioni da eseguire per realizzare la trasformazione, definita nell'algoritmo, dai dati iniziali ai risultati
- # Ogni istruzione dell'algoritmo viene rappresentata all'interno di un **blocco elementare**, la cui forma grafica è determinata dal tipo di istruzione
- # I blocchi sono collegati tra loro da **linee di flusso**, munite di frecce, che indicano il susseguirsi di azioni elementari

I diagrammi a blocchi – 2



Blocchi elementari

I diagrammi a blocchi – 3

- ‡ Un **diagramma a blocchi** è un insieme di blocchi elementari composto da:
- a) un blocco iniziale
 - b) un blocco finale
 - c) un numero finito n ($n \geq 1$) di blocchi di azione e/o di blocchi di lettura/scrittura
 - d) un numero finito m ($m \geq 0$) di blocchi di controllo

I diagrammi a blocchi – 4

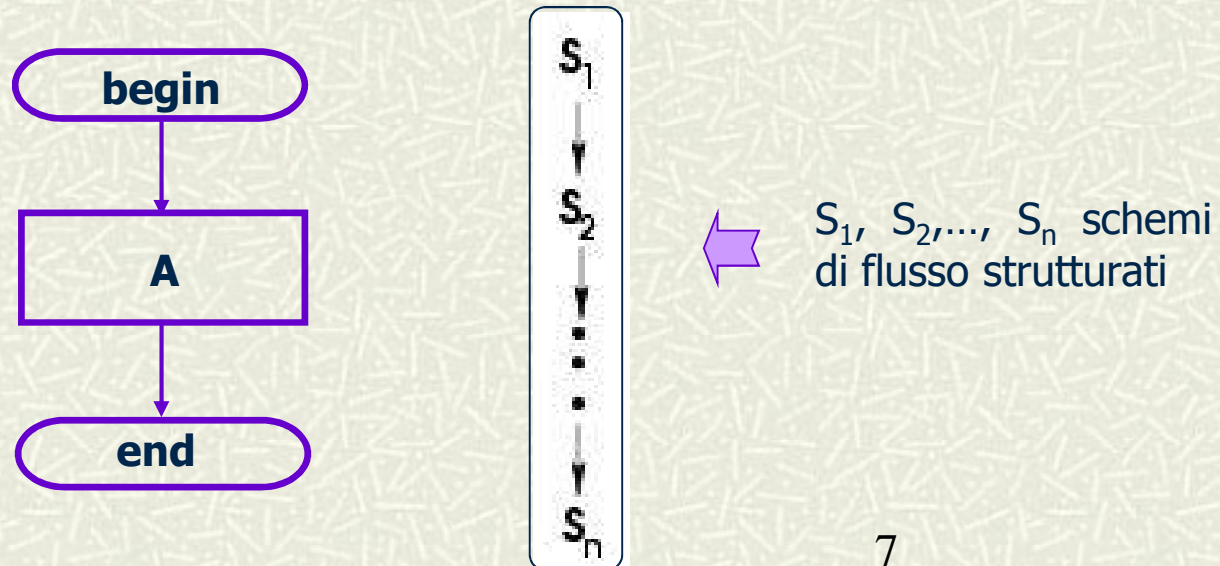
- # L'insieme dei blocchi elementari che descrivono un algoritmo deve soddisfare le seguenti condizioni:
 - ciascun blocco di azione o di lettura/scrittura ha una sola freccia entrante ed una sola freccia uscente
 - ciascun blocco di controllo ha una sola freccia entrante e due frecce uscenti
 - ciascuna freccia entra in un blocco oppure si innesta in un'altra freccia
 - ciascun blocco è raggiungibile dal blocco iniziale
 - il blocco finale è raggiungibile da qualsiasi altro blocco
- # Un blocco B è **raggiungibile** a partire da un blocco A se esiste una sequenza di blocchi X_1, X_2, \dots, X_n , tali che $A=X_1$, $B=X_n$, e $\forall X_i, i=1, \dots, n-1, X_i$ è connesso con una freccia a X_{i+1}

Analisi strutturata – 1

- # I programmatori inesperti tendono ad “aggrovigliare” il programma introducendo numerosi salti privi di regole (*spaghetti programming*)
- # L'**analisi strutturata** favorisce, viceversa, la descrizione di algoritmi facilmente documentabili e comprensibili
- # I blocchi di un diagramma a blocchi strutturato sono collegati secondo i seguenti schemi di flusso:
 - ◆ **Schema di sequenza** – più schemi di flusso sono eseguiti in sequenza
 - ◆ **Schema di selezione** – un blocco di controllo subordina l'esecuzione di due possibili schemi di flusso al verificarsi di una condizione
 - ◆ **Schema di iterazione** – si itera l'esecuzione di un dato schema di flusso

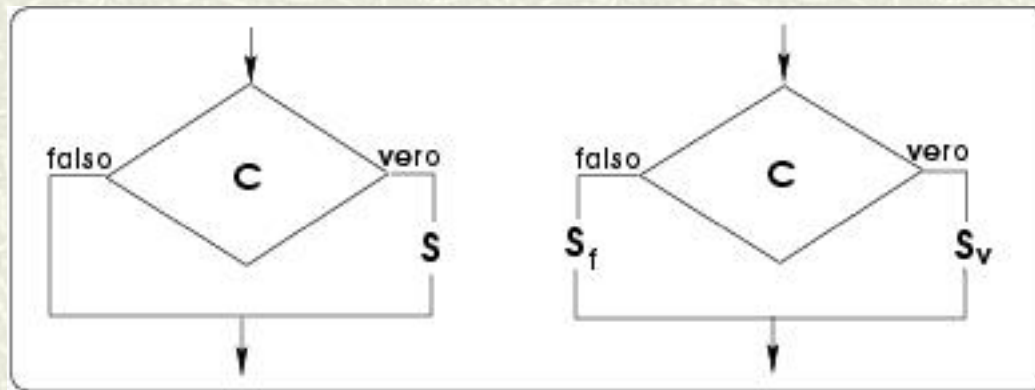
Analisi strutturata – 2

- # Ovvero: un **diagramma a blocchi strutturato** è un diagramma a blocchi nel quale gli schemi di flusso sono strutturati
- # Uno schema di flusso è strutturato quando soddisfa una delle seguenti proprietà...
 - 1) ...è uno schema elementare o uno schema di sequenza



Analisi strutturata – 3

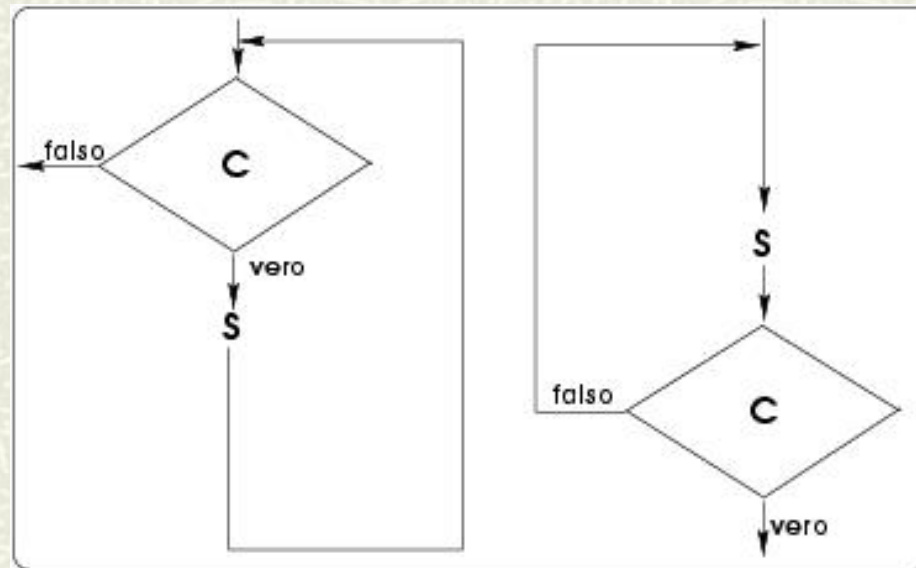
2) ...è uno schema di selezione



- Nel primo caso, lo schema S viene eseguito solo se la condizione C è vera; se C è falsa, non viene eseguita alcuna azione
- Nel secondo caso, viene eseguito solo uno dei due schemi S_v o S_f , in dipendenza del valore di verità della condizione

Analisi strutturata – 4

3) ...è uno schema di iterazione



- Nel primo caso, S può non venire mai eseguito, se la condizione C è subito falsa; nel secondo caso, S viene eseguito almeno una volta
- Quando lo schema S viene eseguito finché la condizione C si mantiene vera si parla di **iterazione per vero**; si ha un'**iterazione per falso** quando S viene eseguito finché C è falsa

Analisi strutturata – 5

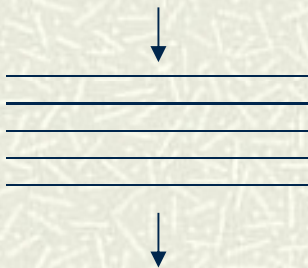
- ✦ Gli schemi di flusso sono **aperti** quando consentono una sola esecuzione di una sequenza di blocchi elementari, sono **chiusi** quando permettono più di un'esecuzione della sequenza di blocchi
- ✦ Gli schemi di sequenza e di selezione sono aperti, lo schema di iterazione è chiuso
- ✦ **Ogni diagramma a blocchi non strutturato è trasformabile in un diagramma a blocchi strutturato equivalente**
- ✦ Due diagrammi a blocchi sono **equivalenti** se, operando sugli stessi dati, producono gli stessi risultati
- ✦ L'uso dell'analisi strutturata garantisce:
 - facilità di comprensione e modifica dei diagrammi a blocchi
 - maggiore uniformità nella descrizione degli algoritmi

Analisi strutturata – 6

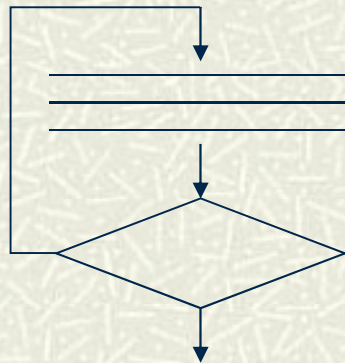
Inoltre...

- ↳ È stato dimostrato (teorema fondamentale della programmazione di Bohm–Jacopini, 1966) che ogni programma può essere codificato riferendosi esclusivamente ad un algoritmo strutturato e quindi attenendosi alle tre strutture fondamentali:

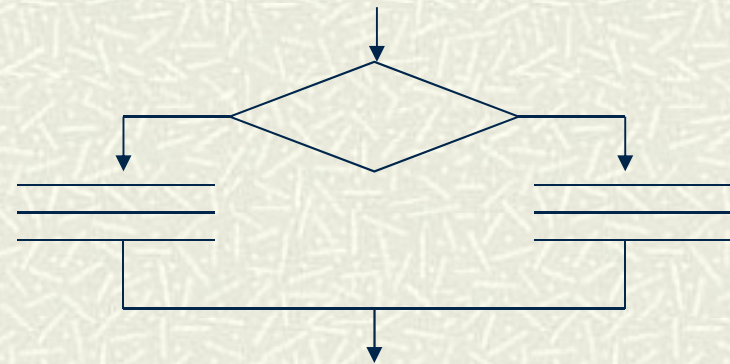
Sequenziale



Iterativa



Condizionale

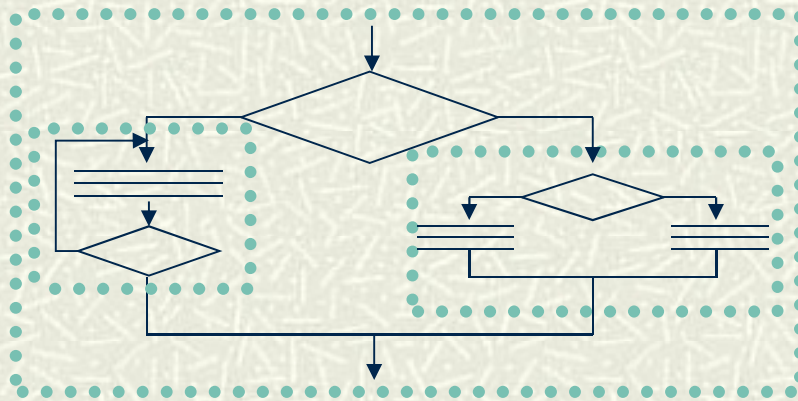


Analisi strutturata – 7

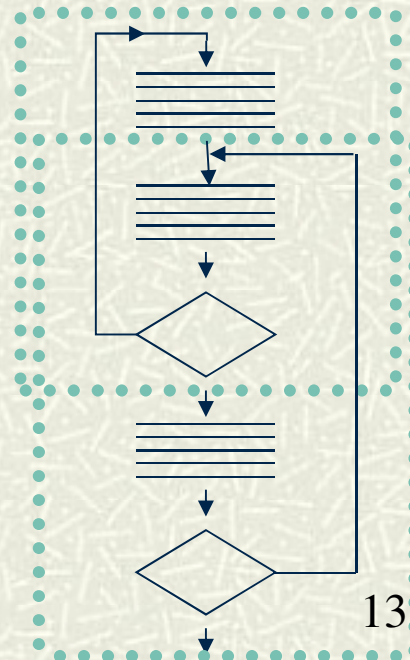
- # Il teorema di Bohm–Jacopini ha un interesse soprattutto teorico, in quanto i linguaggi di programmazione tendono a dotarsi di più tipi di istruzioni, non sempre “rispettose” del teorema, ma utili per la realizzazione di programmi di facile scrittura e comprensione
- # Il suo valore consiste nella capacità di fornire indicazioni generali per le attività di progettazione di nuovi linguaggi e di strategie di programmazione
- # In effetti, esso ha contribuito alla critica dell’uso sconsiderato delle istruzioni *go to* e alla definizione delle linee guida della programmazione strutturata, sviluppate negli anni `70

Analisi strutturata – 8

- In un diagramma strutturato non apparirà mai una istruzione di salto incondizionato
- I tre schemi fondamentali possono essere *concatenati*, uno di seguito all'altro, o *nidificati*, uno dentro l'altro; non possono in nessun caso essere "intrecciati" o "accavallati"



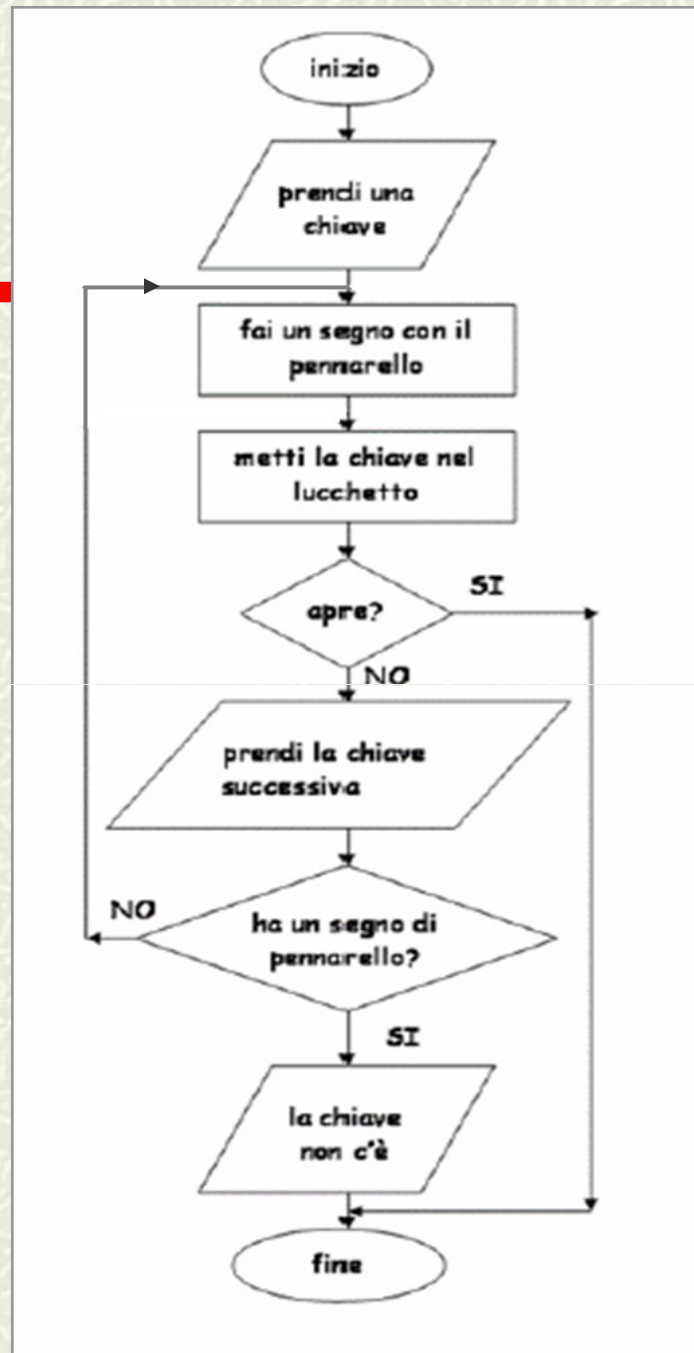
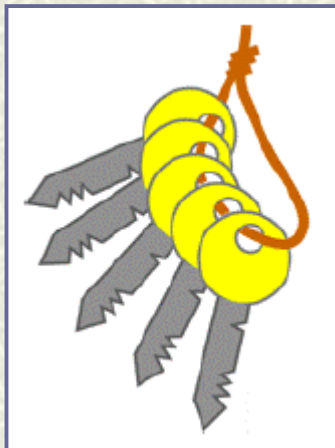
Corretto



Sbagliato

Esempio

- ▣ Diagramma a blocchi per la selezione, in un mazzo di chiavi, di quella che apre un lucchetto

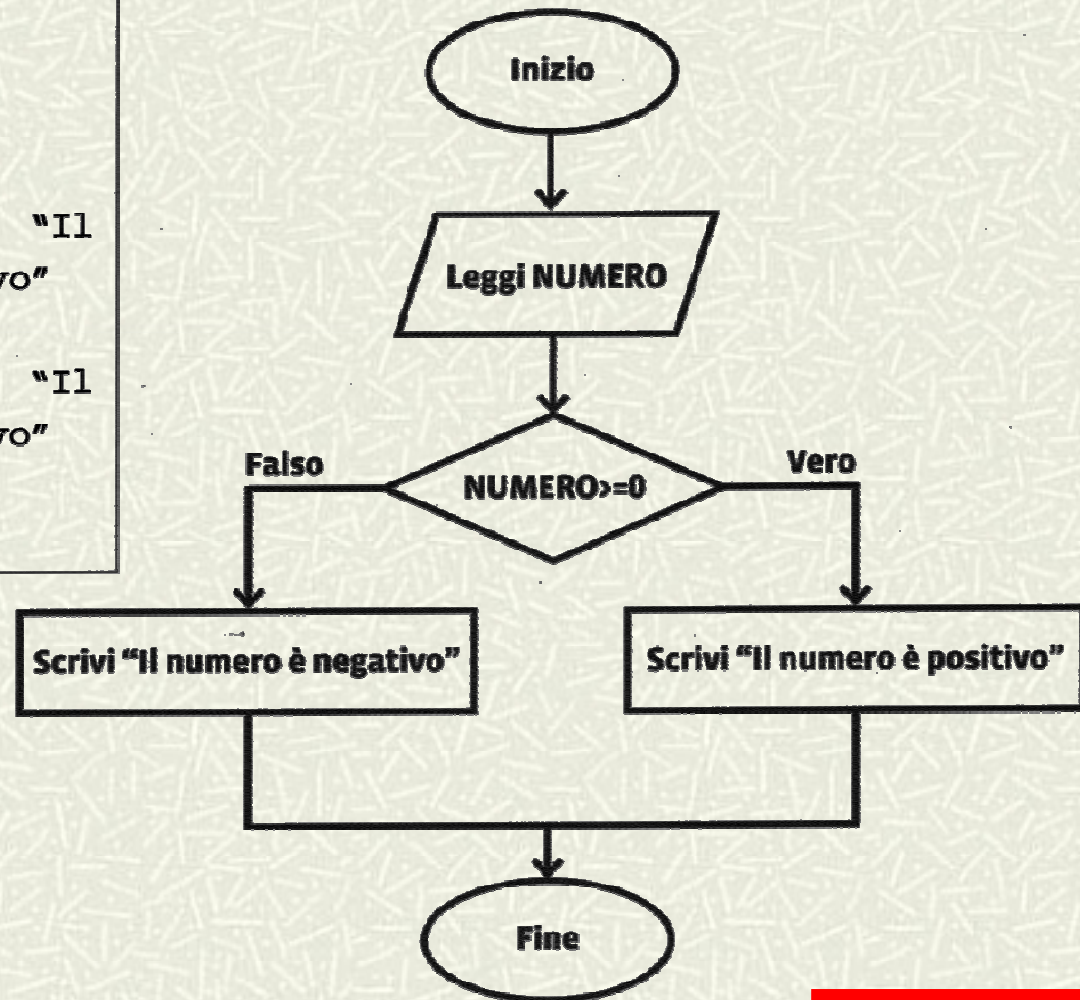


Esempi di analisi

1. Dato un numero dire se è positivo o negativo
 2. ... e se il numero fosse nullo?
 3. Eseguire il prodotto tra due numeri positivi usando solo l'operazione di somma
 4. ... E se i numero non fossero solo positivi?
 5. Trovare il maggiore tra tre numeri
 6. Somma di 3 interi consecutivi
 7. Somma dei primi N numeri
 8. Media di N numeri
 9. Acquisto di prodotti con tetto massimo di spesa
 10. Divisione tra due numeri
-

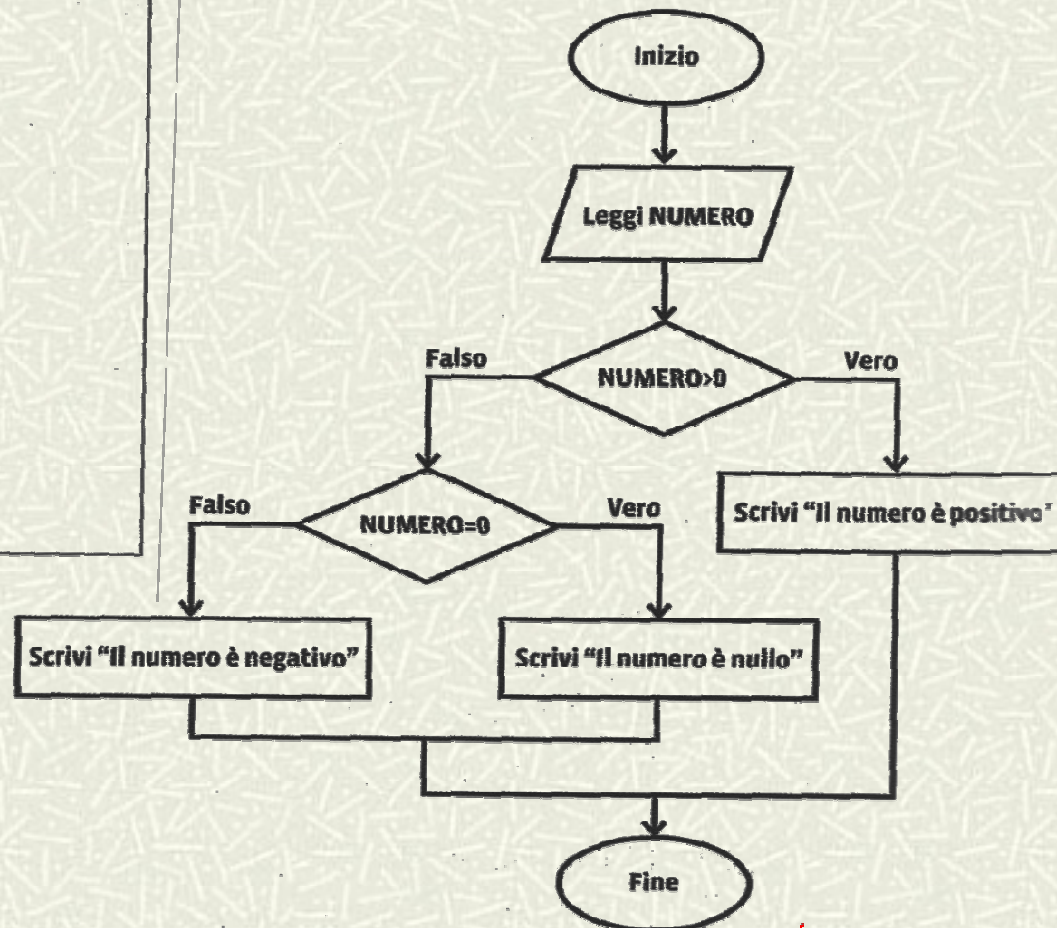
1) Numero positivo o negativo?

```
Inizio
Leggi NUMERO
Se NUMERO >= 0
Allora
  Scrivi "Il
numero è positivo"
Altrimenti
  Scrivi "Il
numero è negativo"
FineSe
Fine
```



2) ... e se fosse nullo?

```
Inizio
Leggi NUMERO
Se NUMERO>0
Allora
  Scrivi "Il numero è positivo"
Altrimenti
  Se NUMERO=0
  Allora
    Scrivi "Il numero è nullo"
  Altrimenti
    Scrivi "Il numero è negativo"
  FineSe
FineSe
Fine
```



3) Prodotto di due numeri

PROBLEMA

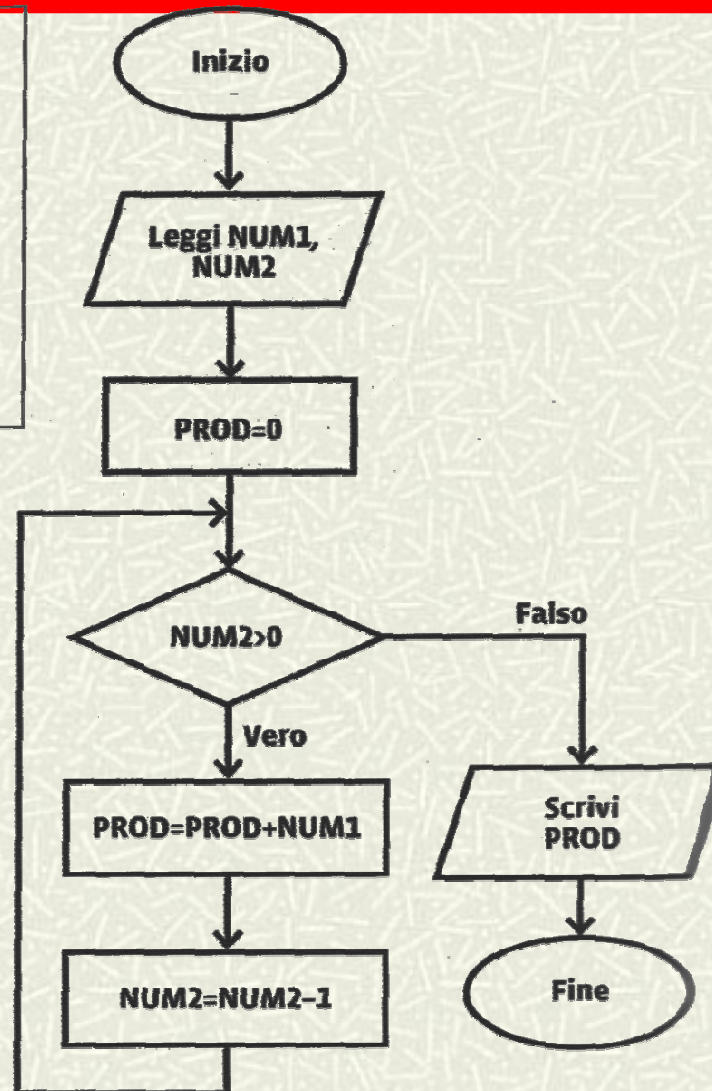
- # Dati due numeri effettuare il loro prodotto usando solo l'operazione somma.

SOLUZIONE

- # Provate a trovare una soluzione al problema descrivendola con diagrammi di flusso

3) Prodotto di due numeri

```
Inizio
Leggi NUM1, NUM2
PROD=0
Mentre NUM2>0
    PROD=PROD+NUM1
    NUM2=NUM2-1
FineMentre
Scrivi PROD
Fine
```



Usando solo
l'operazione
somma

SOLUZIONE:

5) Maggiore tra 3 numeri

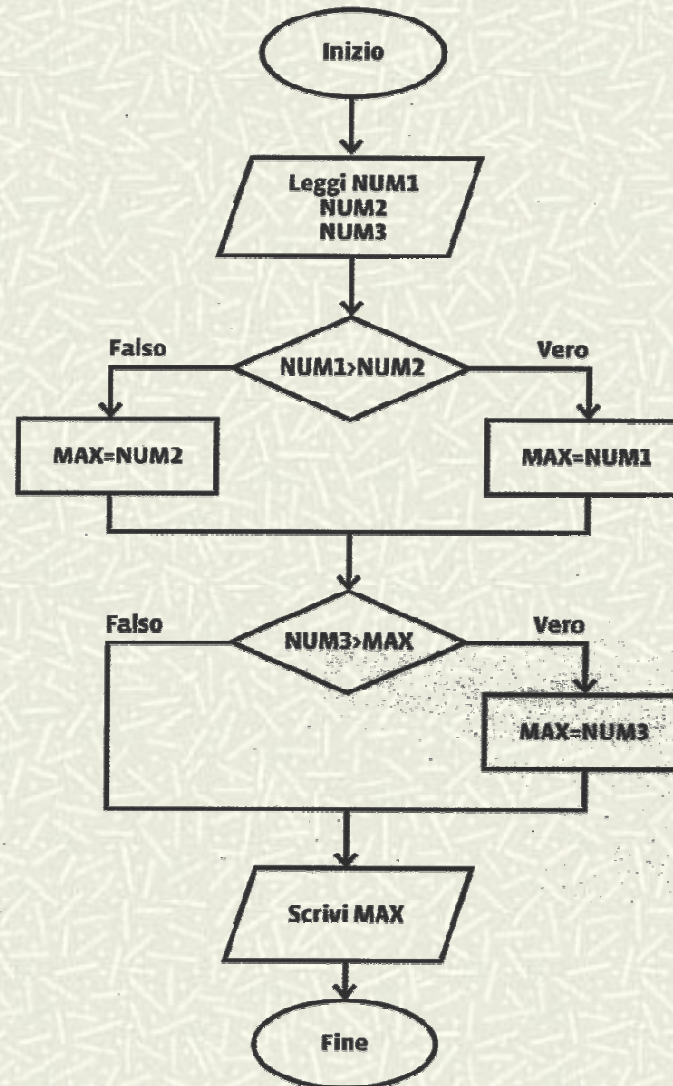
PROBLEMA

- # Dati 3 numeri stabilire quale sia il valore maggiore

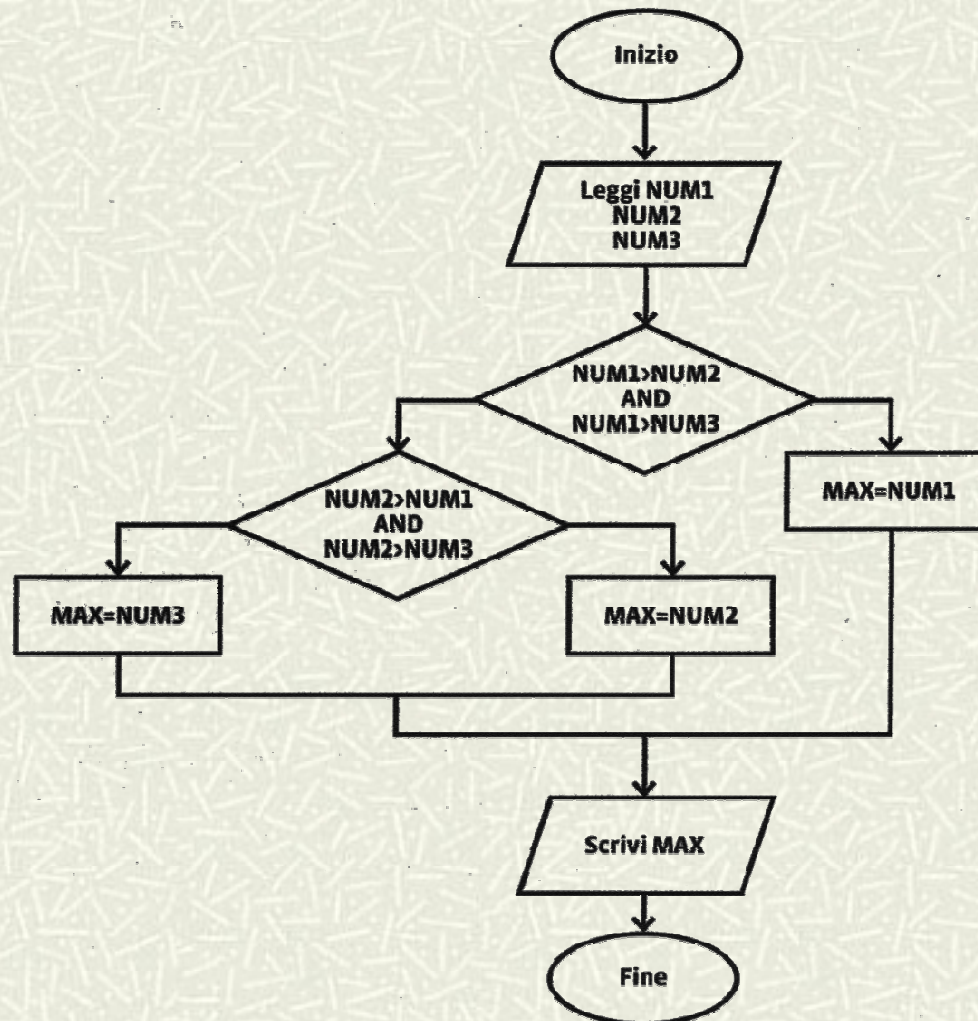
SOLUZIONE

- # Provate a trovare una soluzione al problema descrivendola con diagrammi di flusso

5) Maggiore tra 3 numeri (ver.1)



5) Maggiore tra 3 numeri (ver.2)

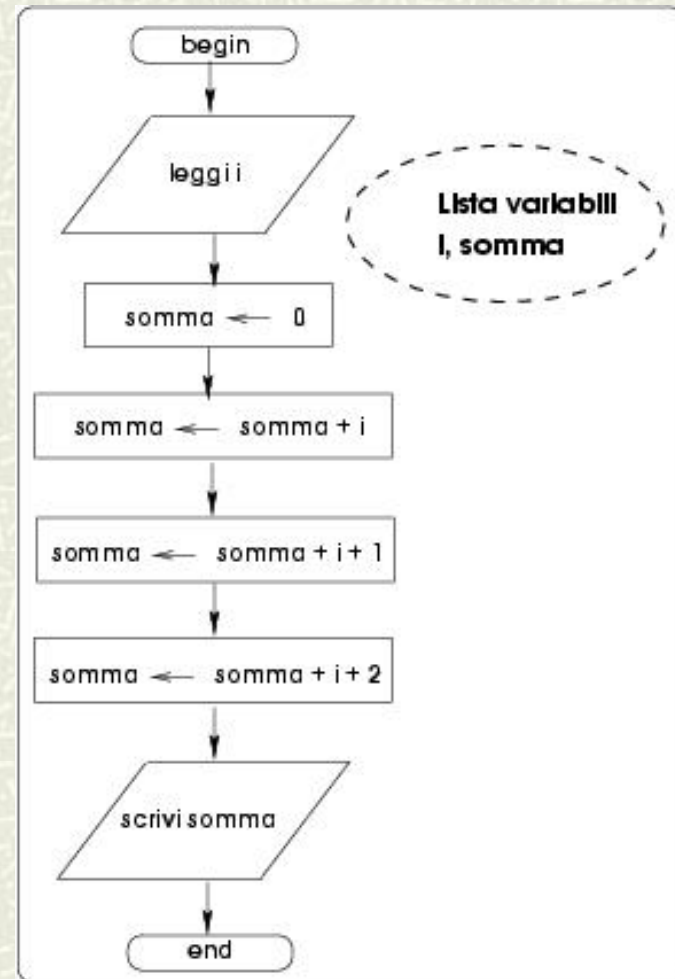


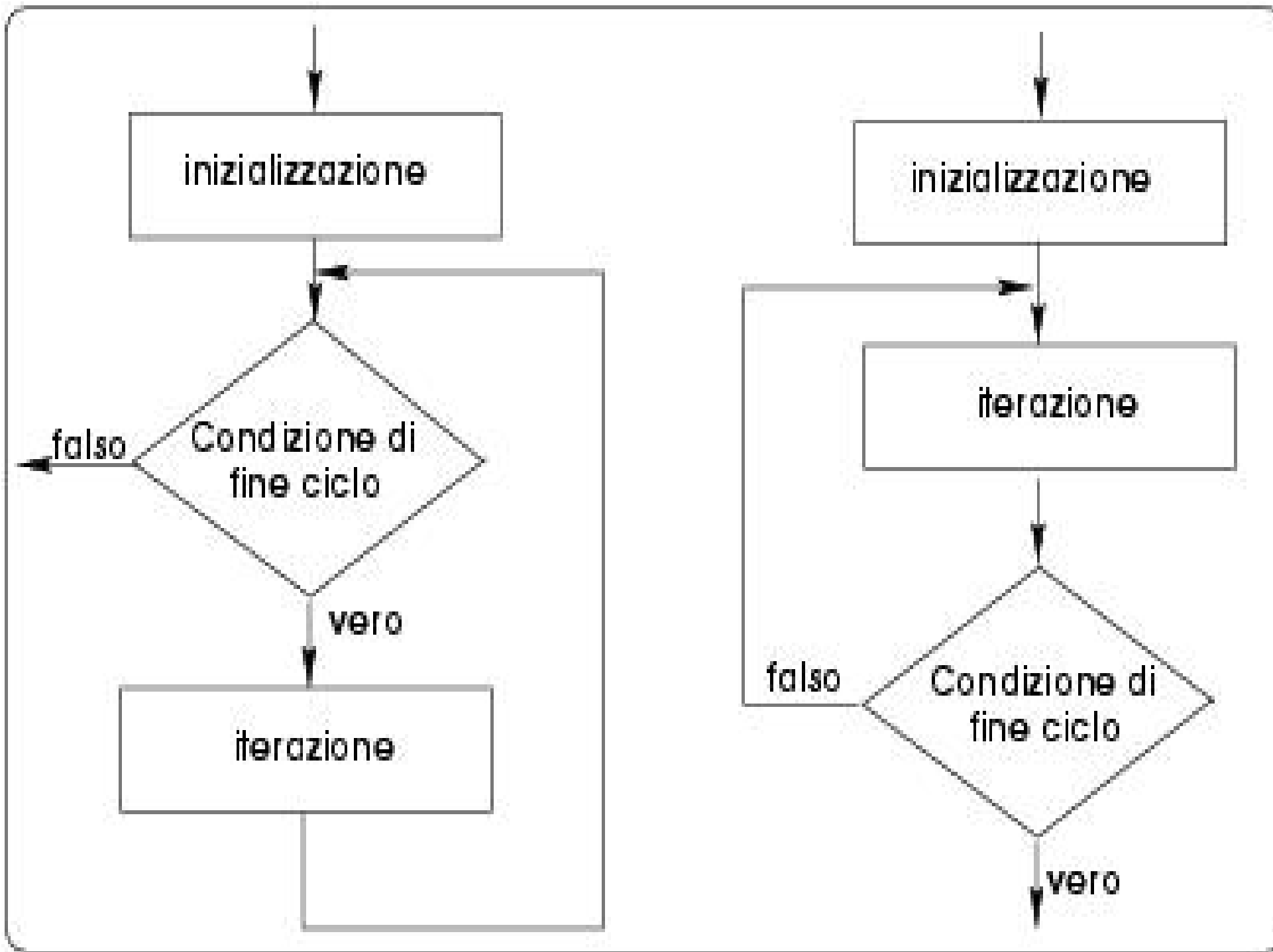
6) Somma di 3 interi consecutivi

- **Problema:** Calcolare la somma di tre interi consecutivi

Note:

- ◆ La variabile *somma* è un contenitore di somme parziali, finché non si ottiene la somma totale richiesta
- ◆ La soluzione del problema viene raggiunta eseguendo azioni simili per un numero opportuno di volte





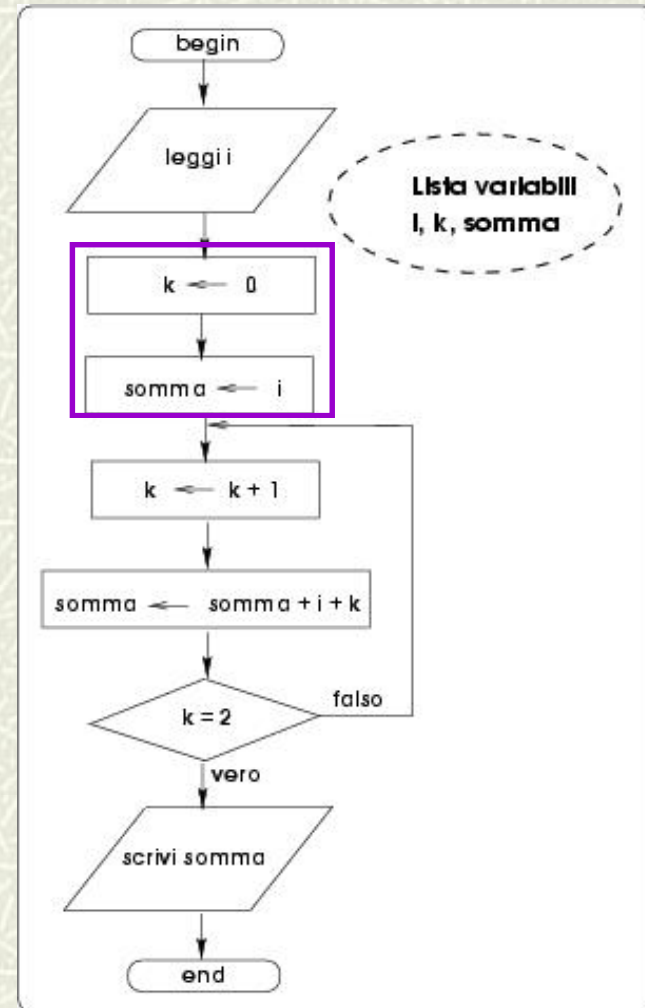
6) Somma di 3 interi consecutivi V.2

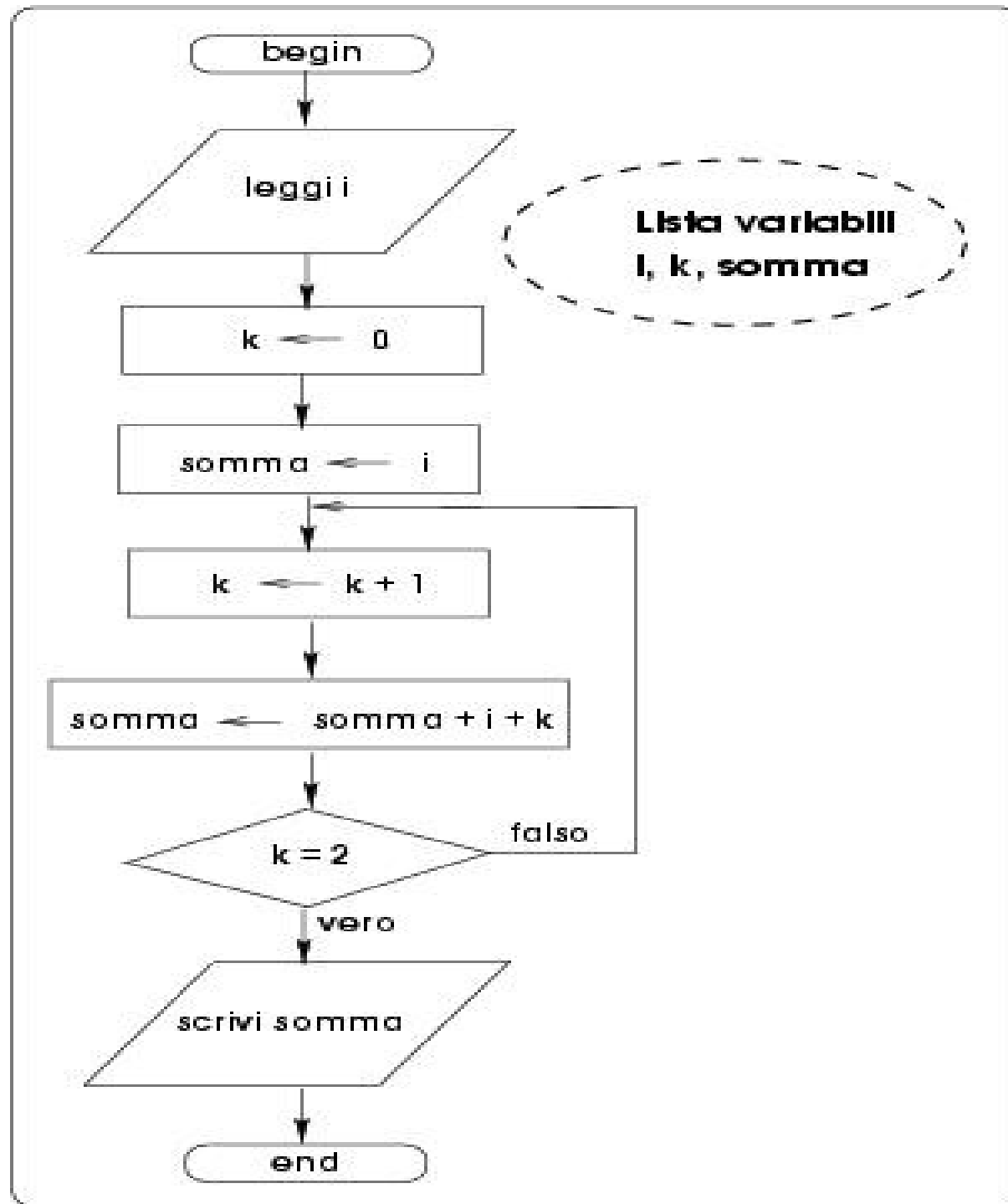
- **Problema:** Calcolare la somma di tre interi consecutivi



Note:

- ◆ La fase di inizializzazione riguarda la somma e l'indice del ciclo
- ◆ Il controllo di fine ciclo viene effettuato in coda





Gli algoritmi iterativi – 2

- # Un ciclo è **definito** quando è noto a priori quante volte deve essere eseguito; un ciclo definito è detto anche **enumerativo**
- # Un **contatore del ciclo** tiene memoria di quante iterazioni sono state effettuate; può essere utilizzato in due modi:
 - incremento del contatore: il contatore viene inizializzato ad un valore minimo (ad es. 0 o 1) e incrementato ad ogni esecuzione del ciclo; si esce dal ciclo quando il valore del contatore eguaglia il numero di iterazioni richieste
 - decremento del contatore: il contatore viene inizializzato al numero di iterazioni richiesto e decrementato di uno ad ogni iterazione; si esce dal ciclo quando il valore del contatore raggiunge 0 (o 1)

Gli algoritmi iterativi – 3

- # Un ciclo è **indefinito** quando non è possibile conoscere a priori quante volte verrà eseguito
- # La condizione di fine ciclo controlla il valore di una o più variabili modificate da istruzioni che fanno parte dell'iterazione
- # Comunque, un ciclo deve essere eseguito un numero finito di volte, cioè si deve sempre verificare la terminazione dell'esecuzione del ciclo

7) Somma dei primi N numeri

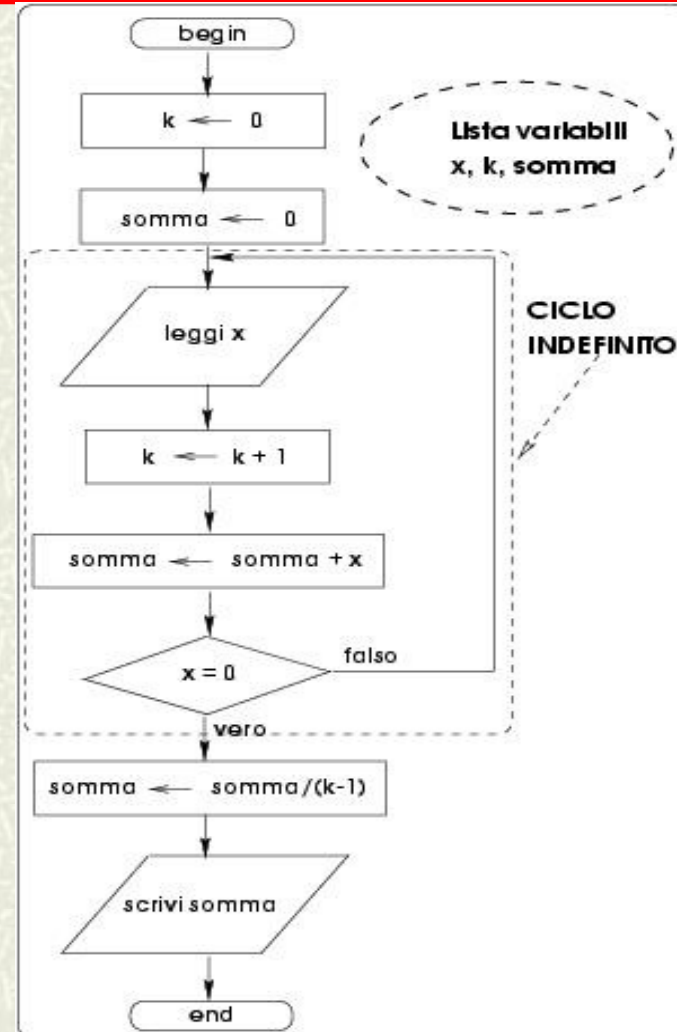
- **Problema:** Calcolare la somma dei primi N numeri.

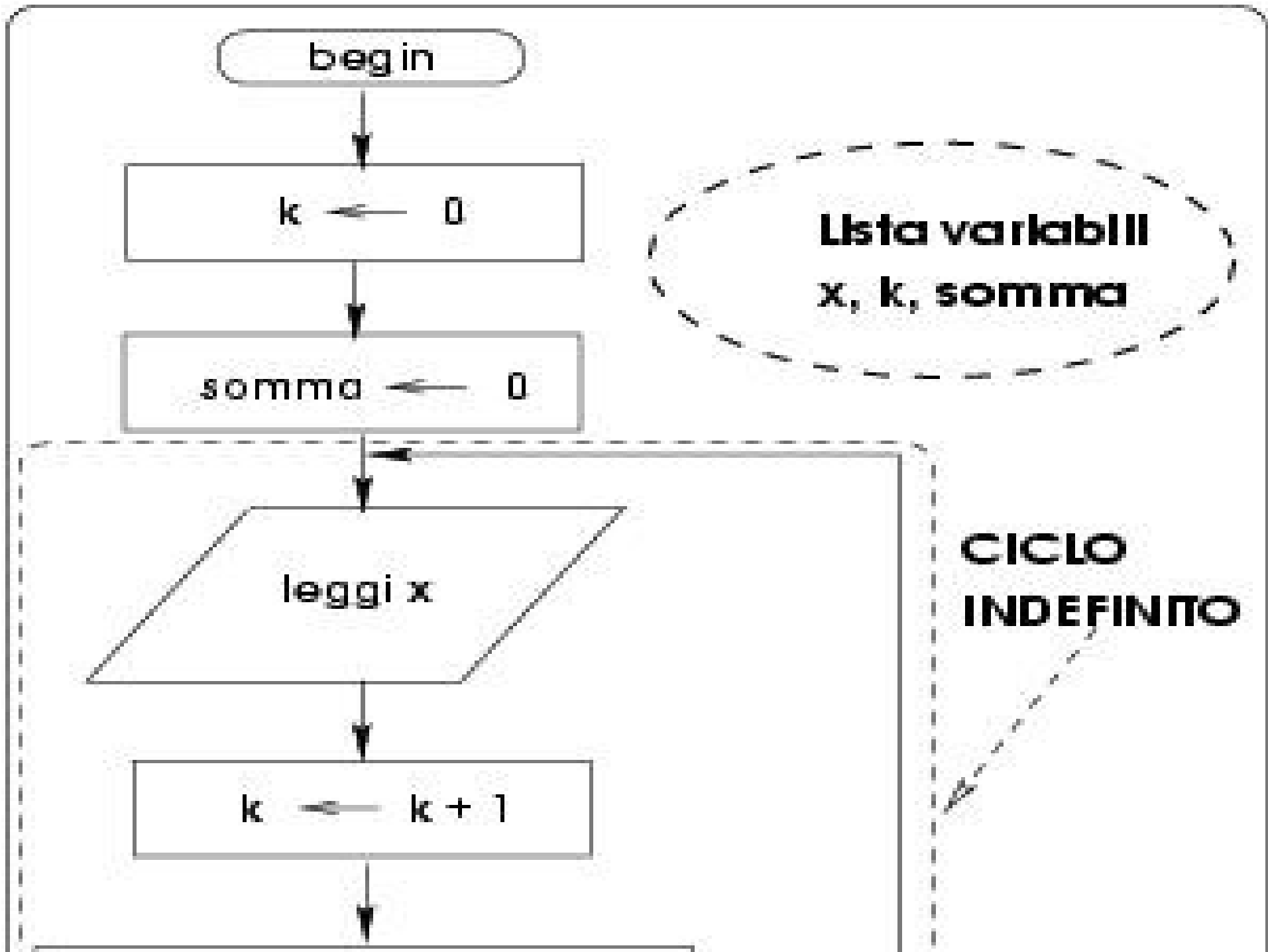
Note:

- ◆ Leggo un numero N e sommo tutti i numeri dell'intervallo $[0,N]$ oppure $[N,0]$ se N è negativo
- ◆ Attenzione al segno del risultato

8) Media di N numeri

- **Problema:** Calcolo della media di un insieme di numeri; non è noto a priori quanti sono i numeri di cui deve essere calcolata la media
 - I numeri vengono letti uno alla volta fino a che non si incontra un $x=0$, che segnala la fine dell'insieme





9) Spesa di prodotti per max 100€

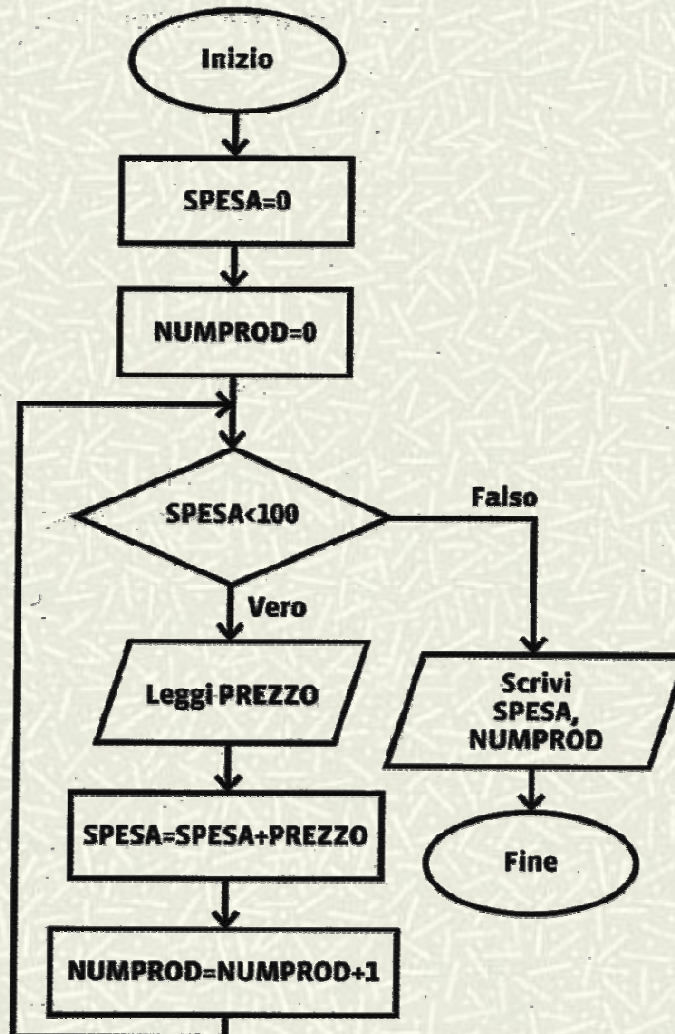
PROBLEMA

- # Si progetti un terminale per automatizzare la spesa in un supermercato. Il terminale deve contare il numero di prodotti messi nel carrello per una spesa massima di 100€.

SOLUZIONE

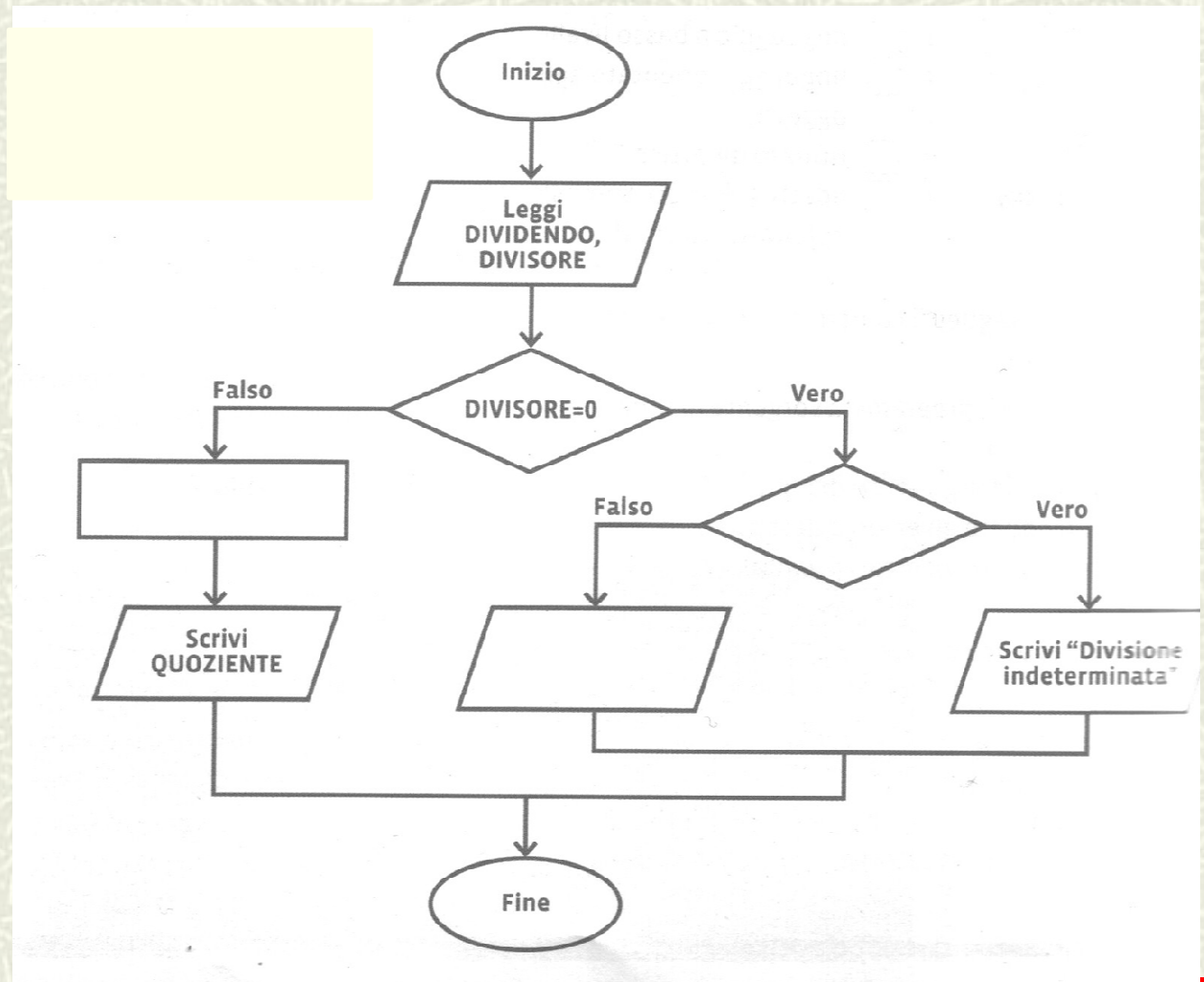
- # Analizzare il problema e descrivere la soluzione con diagrammi di flusso
-

9) Spesa di prodotti per max 100€



10) Divisione tra due numeri (per casa)

- # Completa il diagramma
- # Verifica se è impossibile (divisore=0) o indeterminata (dividendo=0 e divisore=0)



Esercizi per Casa

- # Scrivere un algoritmo, e rappresentarlo tramite diagramma a blocchi, per la soluzione dei seguenti problemi:
 - calcolare l'area del triangolo
 - trovare il max di due numeri
- # Formalizzare, tramite diagramma a blocchi, l'algoritmo per...
 - ...calcolare le radici reali di equazioni di 2° grado
 - ...calcolare il M.C.D. di due numeri con il metodo di Euclide