

# COSTRUTTI DI PROGRAMMAZIONE

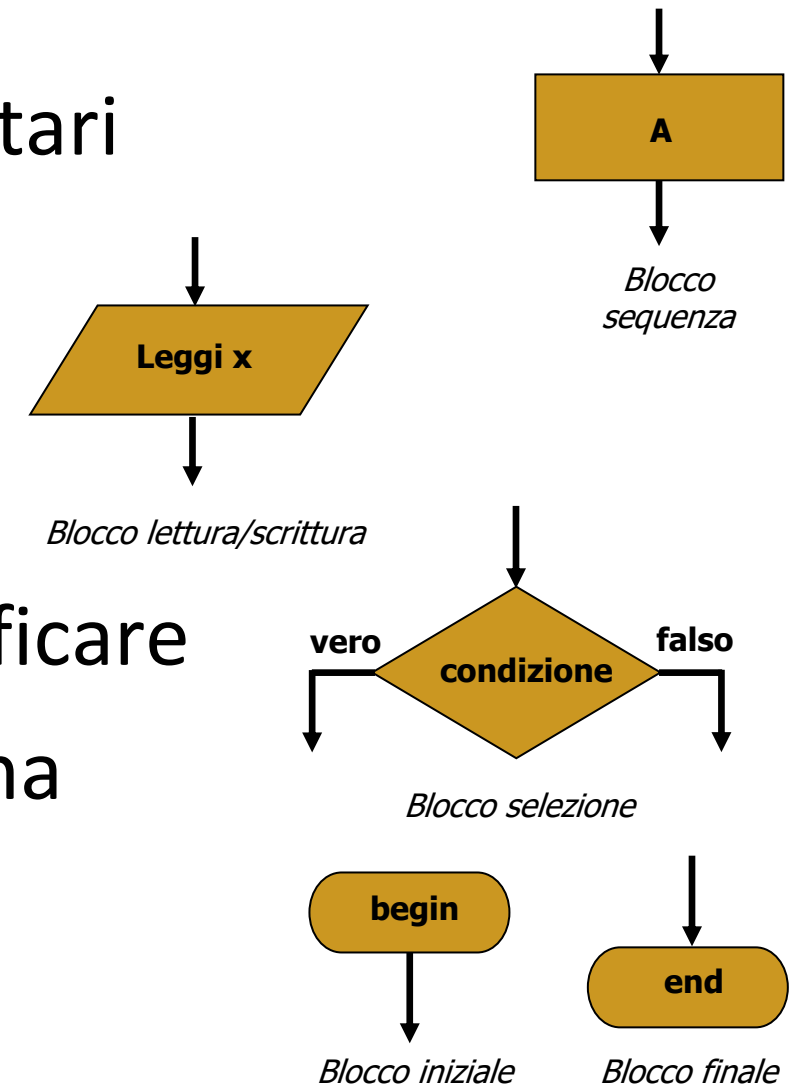
---

Fondamenti di Programmazione 2022/2023



# Blocchi elementari del diagramma di flusso

- Abbiamo già introdotto i blocchi elementari
- **Rettangolo** → assegnazione/operazione
- **Parallelogramma** → un'operazione di input/output
- **Rombo** → selezione : condizione da verificare
- **Ovale** → l'inizio o la fine di un programma



# I costrutti di programmazione

---

- Nella maggior parte dei linguaggi di programmazione, le strutture di controllo (costrutti) sono impiegate in un programma per controllarne il **flusso di esecuzione**;
- Il flusso di esecuzione serve a specificare:
  - **se, quando, in quale ordine e quante volte** devono essere eseguite le istruzioni che compongono il programma stesso.
- I costrutti sono strumenti per costruire (da cui il nome costruito) istruzioni composte a partire dalle istruzioni semplici (assegnazione e lettura/scrittura).



# I costrutti di programmazione

---

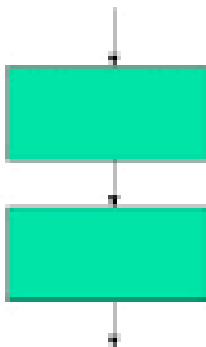
- Sebbene ogni linguaggio di programmazione ha il proprio insieme di costrutti che può differire da quello impiegato da un altro linguaggio, si evincono alcuni **requisiti comuni nella programmazione strutturata**:
  - le strutture sono del tipo **one-in one-out** (singolo ingresso, singola uscita); sono considerate come una singola macro-istruzione, con un unico e ben identificato punto di ingresso e un unico punto di uscita;
  - le strutture di controllo possono contenere al loro interno istruzioni a loro volta composte, senza limiti a tale procedimento di **nesting** (annidamento)
  - l'insieme delle strutture di controllo è funzionalmente completo: sono cioè sufficienti a scrivere un qualsiasi algoritmo → **teorema di Bohm e Jacopini**



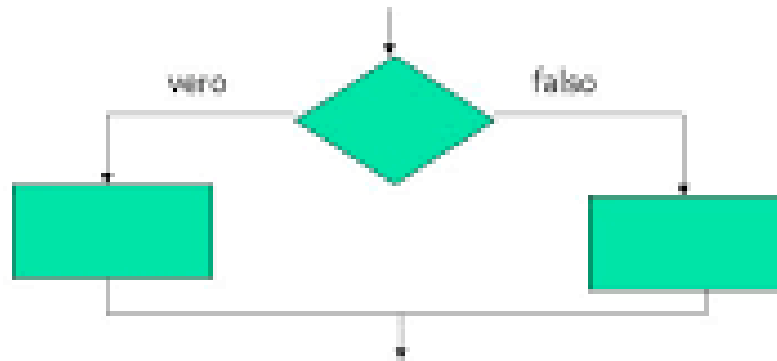
# Teorema di Böhm-Jacopini

- Il teorema di Böhm-Jacopini (1966) afferma che:  
***“qualunque algoritmo può essere implementato utilizzando tre sole strutture, la sequenza, la selezione e il ciclo (o iterazione), da applicare ricorsivamente alla composizione di istruzioni elementari”.***

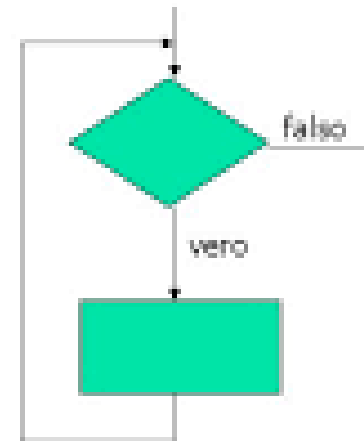
Sequenza



Selezione



Iterazione



# I costrutti di programmazione

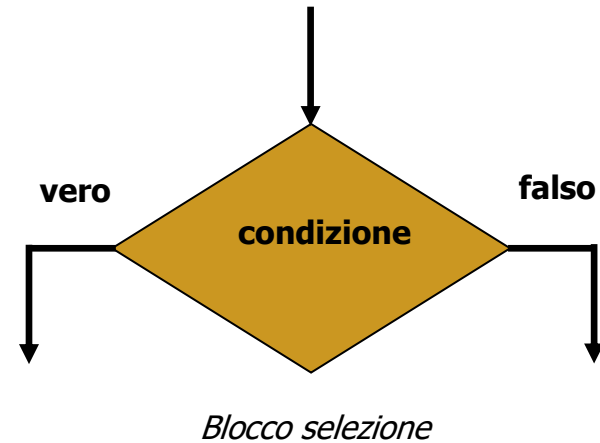
---

- Tre tipi di costrutti di controllo:
  - Sequenza
  - Costrutti selettivi : consentono di operare delle scelte
  - Costrutti iterativi: consentono di ripetere alcune operazioni un certo numero di volte
- Mediante l'uso dei flow-chart, è possibile combinare i blocchi per realizzare i costrutti selettivi e iterativi



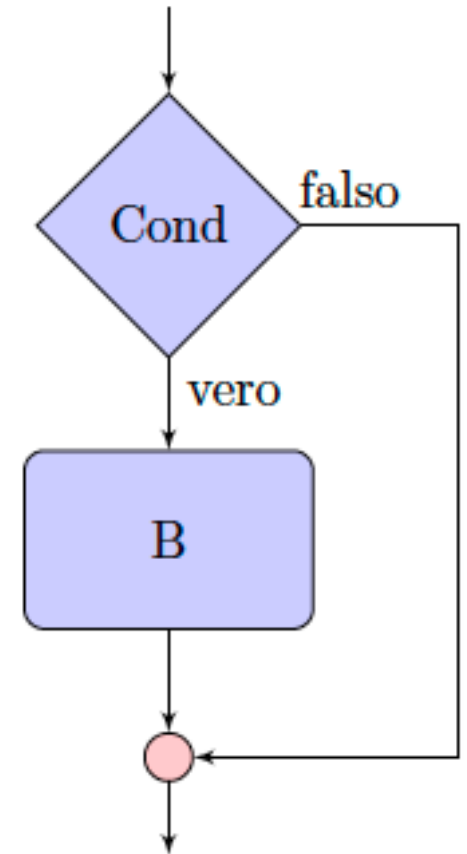
# Costrutti selettivi

- I costrutti selettivi (dette anche strutture di controllo) consentono di instradare il flusso di controllo dell'algoritmo su più strade in funzione del valore assunto da una condizione specificata nel rombo.
- Esistono diverse tipologie di costrutti selettivi che si differenziano in base al numero di strade differenti previste e alla natura della condizione.
- I più semplici sono: **if-then** e **if-then-else**



# Costrutto selettivo if-then

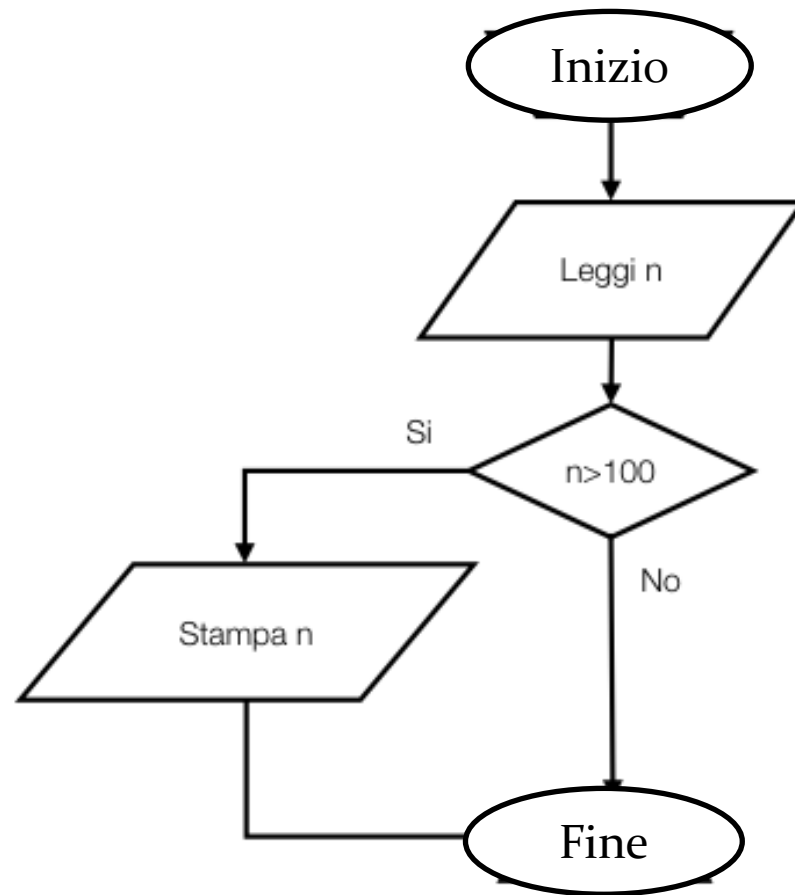
- Il costrutto if-then è la tipologia più semplice di selezione e viene detta selezione unaria ( o selezione ad una via);
  - Notare: solo su una «via» si eseguono le istruzioni!
- Controlla una condizione (**Cond**) che, se verificata, abilita l'esecuzione di un **blocco sequenza B** (o blocco istruzioni, con una o più istruzioni)
- La **Cond** è un'espressione logica che può assumere uno dei due valori:
  - 1, 0
  - Vero, Falso
  - Si, No





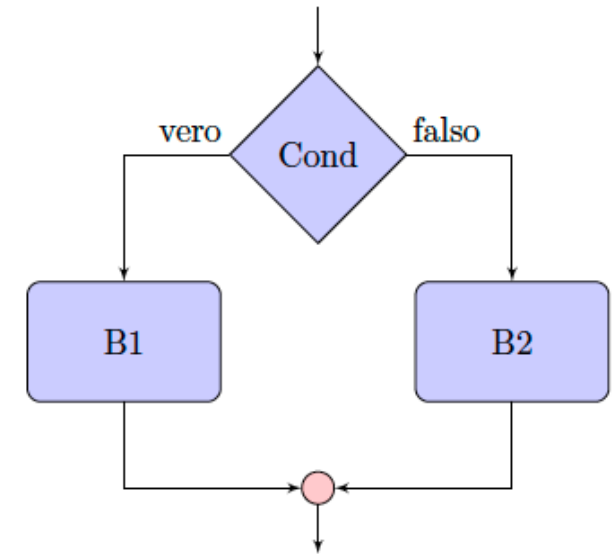
# Cosa fa l'algoritmo?

---



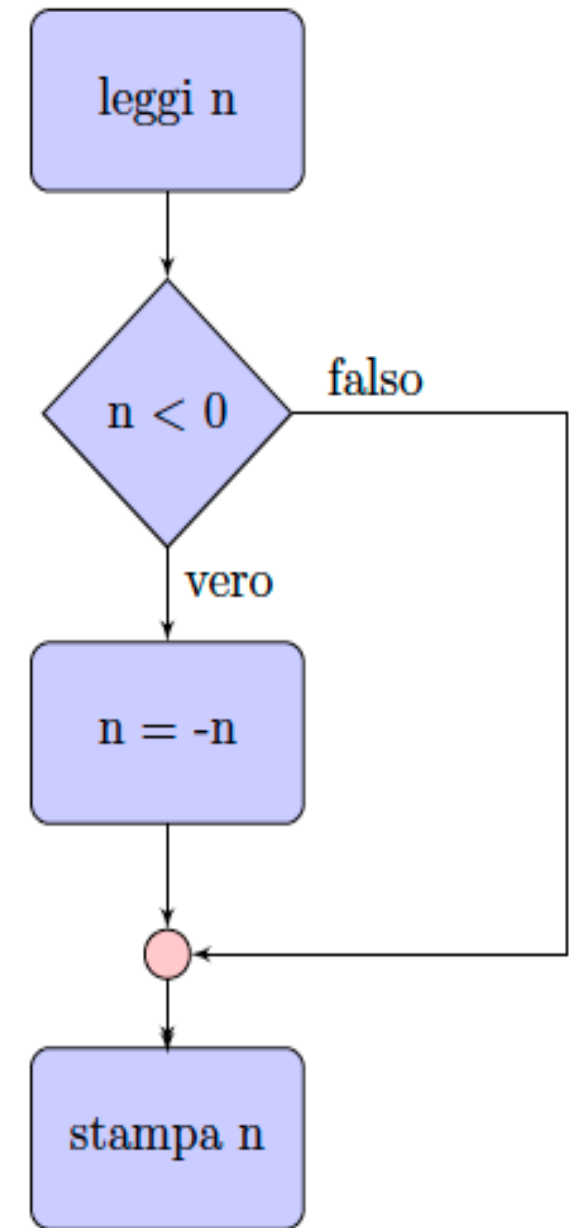
# Costrutto selettivo - if-then-else

- Il costrutto *if-then-else* è detto selezione binaria (o a due vie);
- Prevede l'esecuzione di due blocchi alternativi di istruzioni, sulla base del risultato della condizione **Cond**.
- Se **Cond** assume il valore vero, viene eseguito il blocco di istruzioni B1, altrimenti (cioè, se **Cond** è falsa) viene eseguito il blocco B2.
- Come per la selezione unaria, la selezione a due vie prevede che la condizione **Cond** sia di tipo binario, ovvero tale da assumere uno tra i due valori possibili, *vero o falso*.



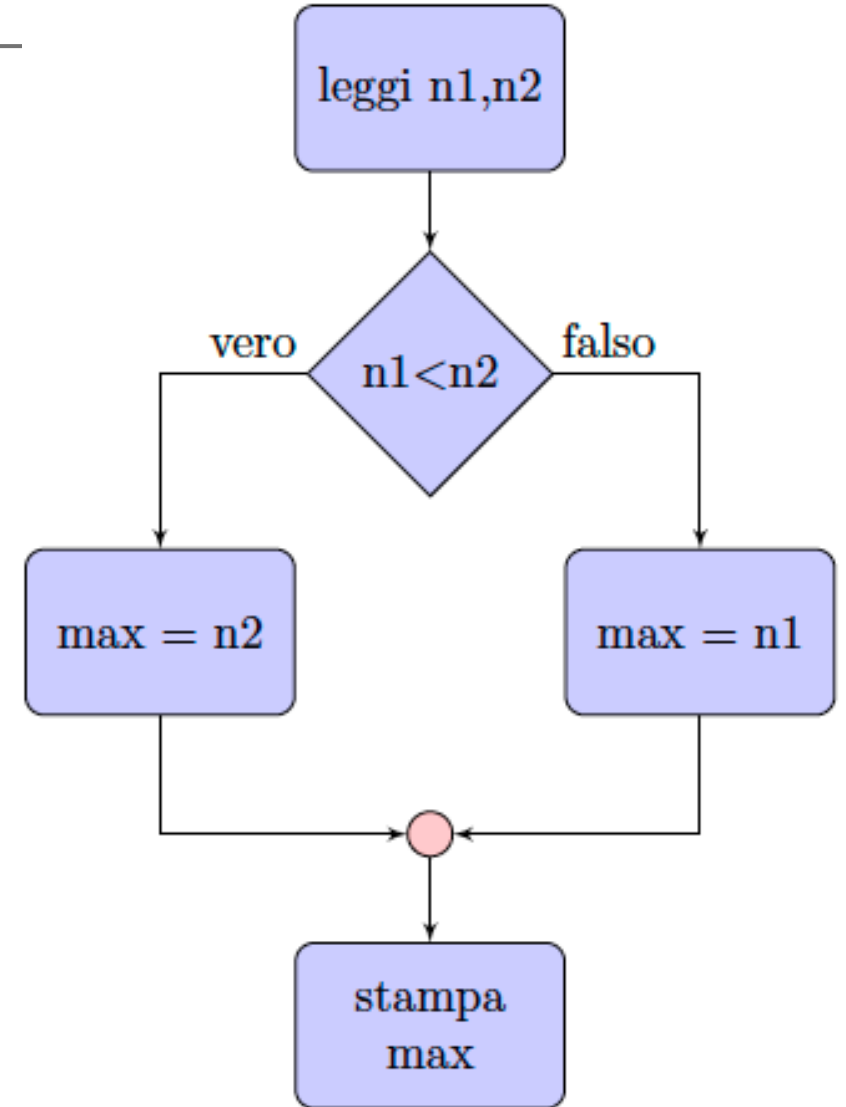
# Costrutto selettivo

- Esempio: Dato un numero intero  $n$ , se ne valuti il modulo (o valore assoluto).



# Costrutto selettivo

- Esempio: Calcola il massimo tra due numeri in ingresso



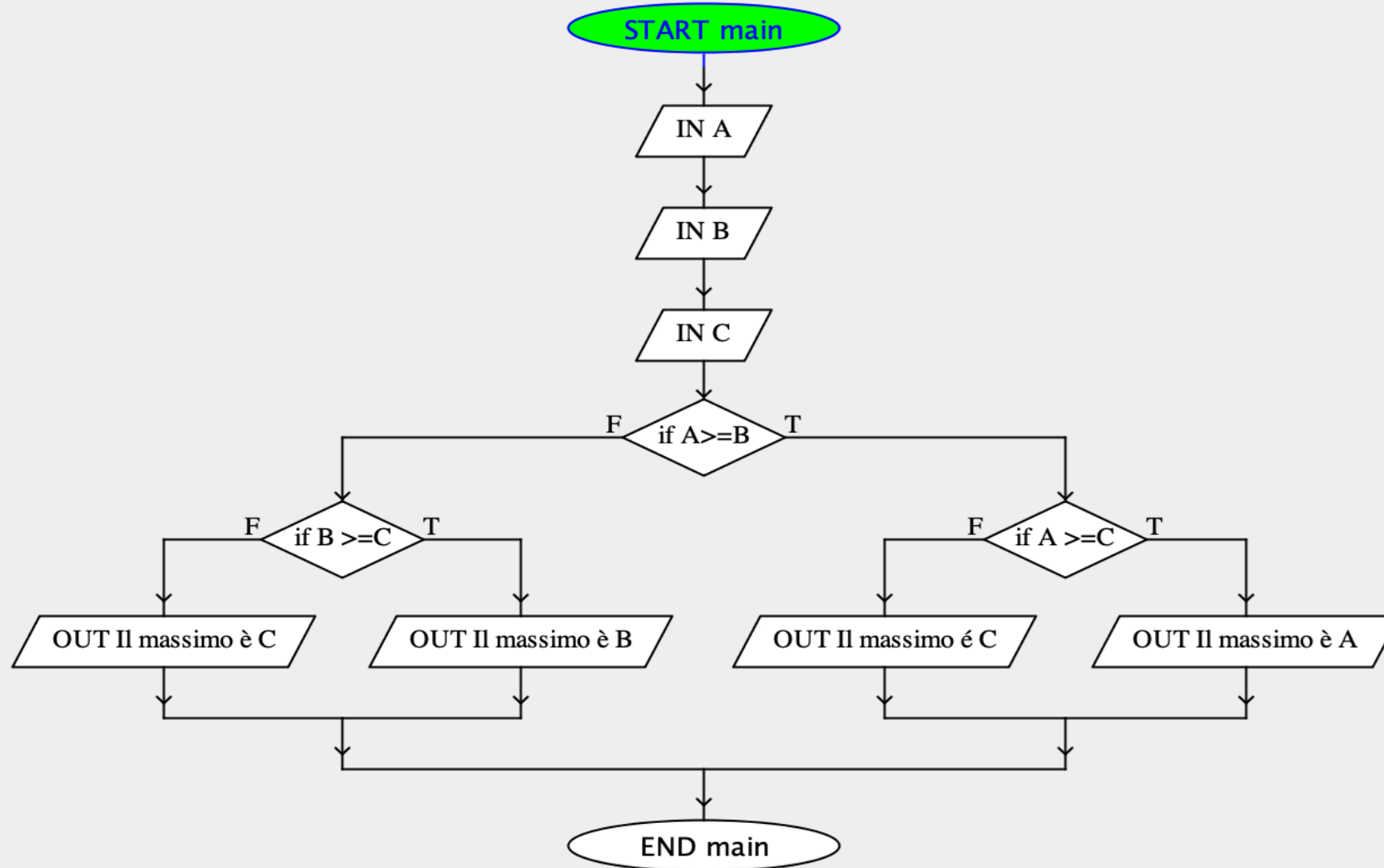
# Esercizio

---

- Calcolare il massimo tra tre interi dati in input.

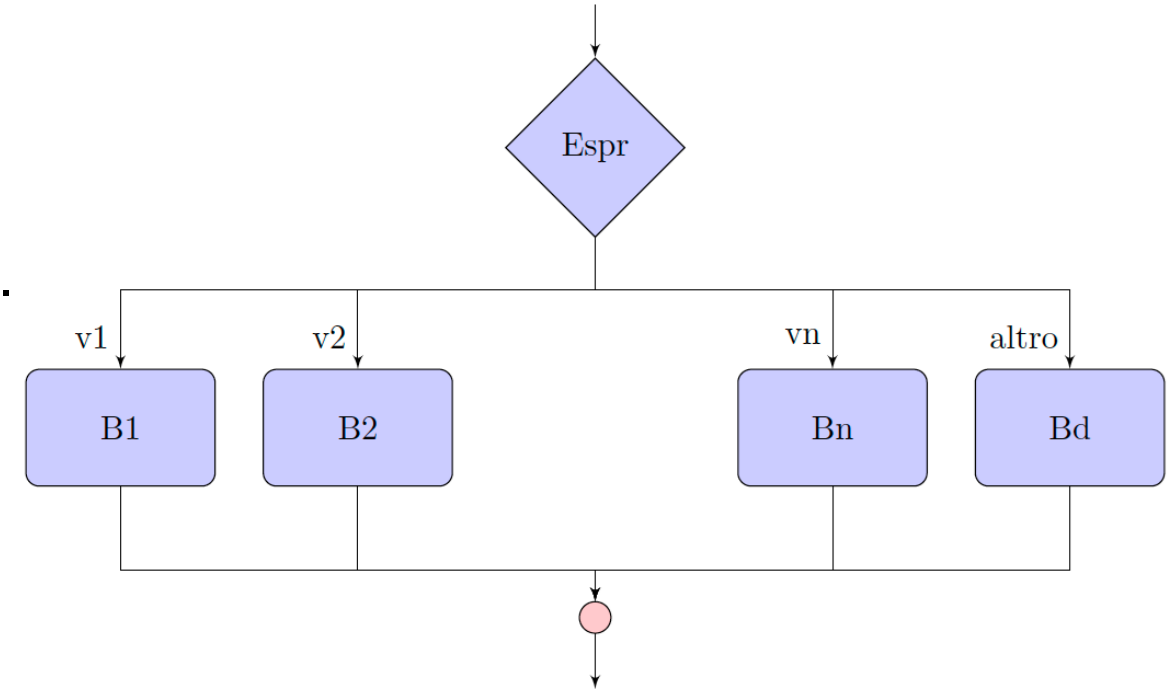


# Possibile Soluzione



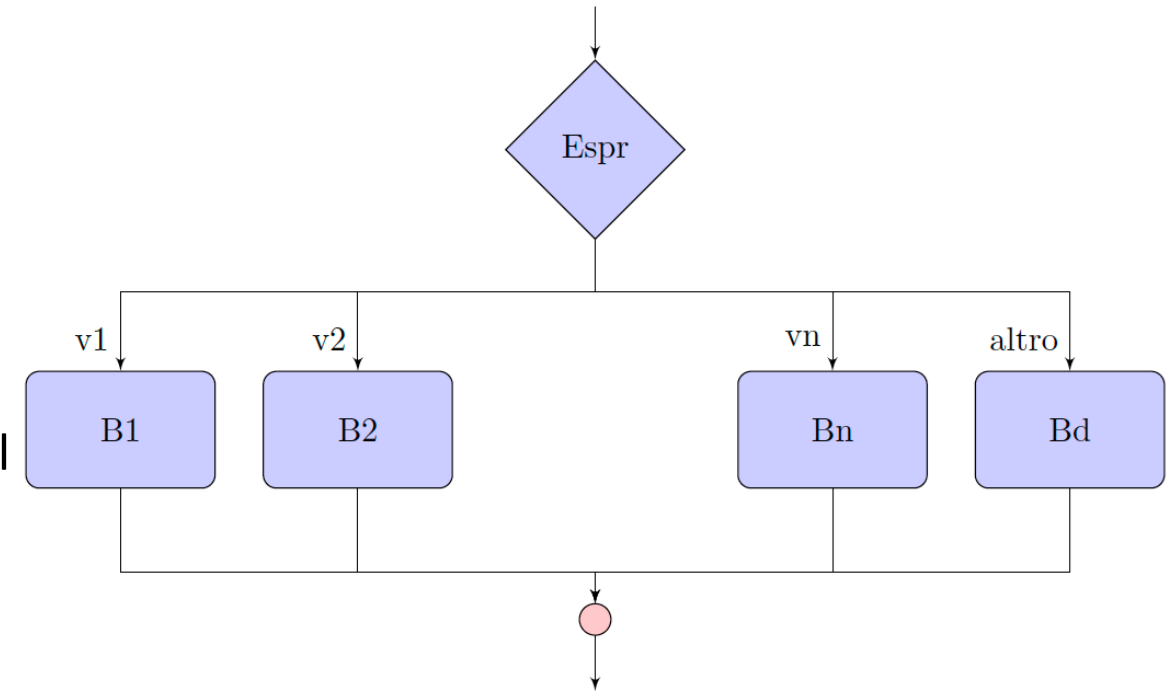
# Costrutto selettivo: case

- Il case è una generalizzazione del costrutto selettivo in cui siano presenti più di due strade distinte.
- La sua logica di funzionamento è molto semplice.
  - Si hanno  $n$  strade ( $B1, \dots Bn$ ), ognuna delle quali associata a uno dei possibili valori assunti dall'espressione  $Espr$  (detta espressione di selezione);
  - C'è un'ulteriore strada ( $Bd$ ), detta di default, che viene percorsa allorquando il valore assunto dall'espressione di selezione non coincide con nessuno dei valori associati alle  $n$  strade.
  - $Espr \text{ NON}$  è una condizione logica ma un'espressione



# Costrutto selettivo: case

- *Il case è una generalizzazione del costrutto selettivo in cui siano presenti più di due strade distinte.*
- *La sua logica di funzionamento è molto semplice.*
- **NOTARE:**
  - se la variabile assume il valore v1 viene eseguito il blocco B1 e poi si esce dal costrutto,
  - se assume il valore v2 il blocco B2 e poi si
  - esce,
  - ...ecc.
  - Il blocco Bd viene percorso se la variabile assume valore diverso da v1, v2, vn.





# Costrutti iterativi

---

- Le strutture di controllo iterative, dette cicli, consentono di eseguire ripetutamente una data istruzione (o un blocco di istruzioni).
- Ogni costrutto iterativo deve consentire di specificare sotto quali condizioni l'iterazione (ripetizione) di tali istruzioni debba terminare → condizione di permanenza nel ciclo.



# Costrutti iterativi

---

- Elementi di un costrutto iterativo:
  - **Inizializzazione**: le variabili usate, e soprattutto quelle nella condizione del ciclo, devono avere un valore iniziale.
  - **Condizione**: deve essere valutata, per determinare la sua ripetizione o la terminazione del ciclo.
  - **Modifica**: almeno una delle variabili della condizione deve essere modificata all'interno del ciclo, in modo che prima o poi la condizione di ripetizione diventi falsa, causando la **terminazione del ciclo**.



# Costrutti iterativi

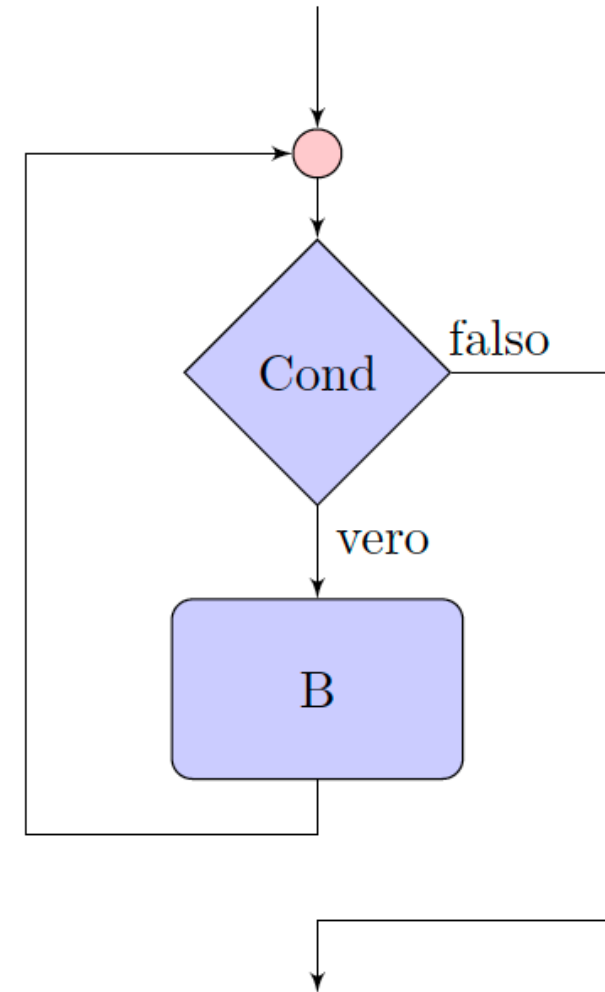
---

- Una prima classificazione dei costrutti iterativi si può ottenere in base al numero delle iterazioni:
  - **predeterminato**: noto prima di entrare nel ciclo stesso
  - **non predeterminato**: il numero delle iterazioni può cambiare durante l'esecuzione del ciclo
- I cicli iterativi sono:
- Il ciclo **for**, il **while**, **do-while** e il **repeat-until**



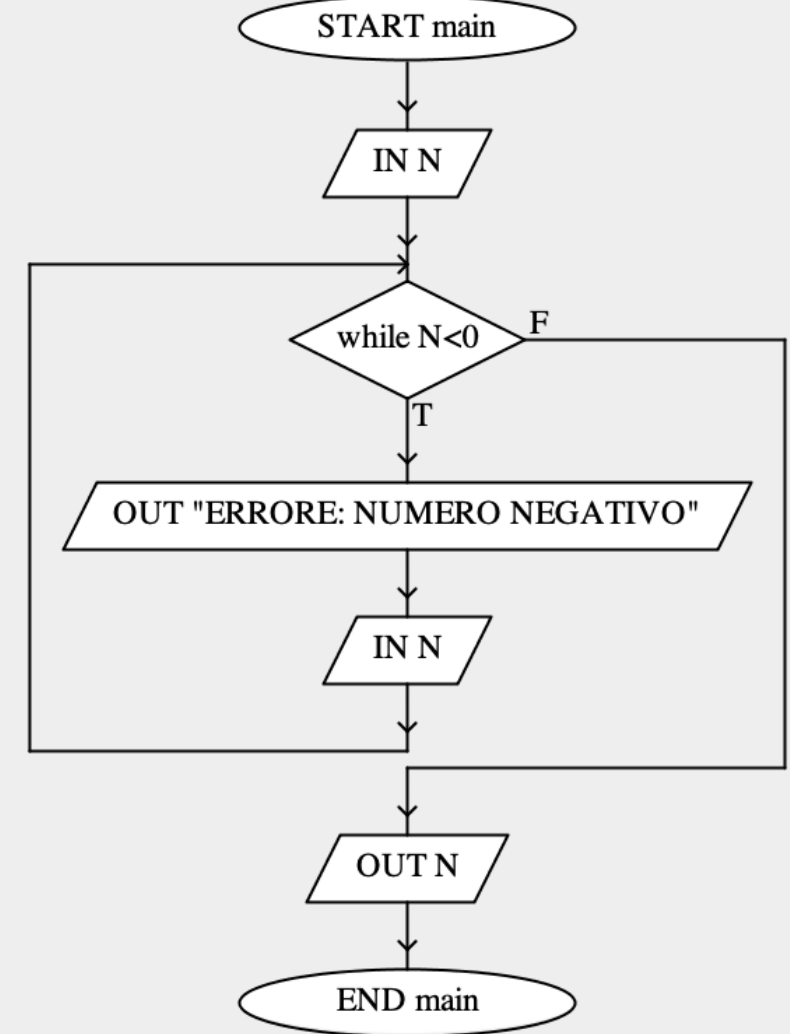
# Ciclo **while**

- Il ciclo **while** è un costrutto iterativo in cui la ripetizione è espressa in termini di una condizione logica che indica «fin quando si continua la ripetizione» (permanenza nel ciclo):
- cioè se **Cond** è vera il ciclo continua a ripetere le istruzioni (blocco sequenza B) in esso contenuto.
- Detto anche ciclo a condizione iniziale
- Le forme tradizionali di tale ciclo possono così essere parafrasate: **finchè **Cond** è verificata, ripeti B**



# Ciclo

- **Esercizio**
- Dato un un numero intero, verificare se è positivo; in caso non lo sia, stampare un messaggio di errore e ripete l'inserimento.
- Una volta inserito un valore valido, stamparlo.

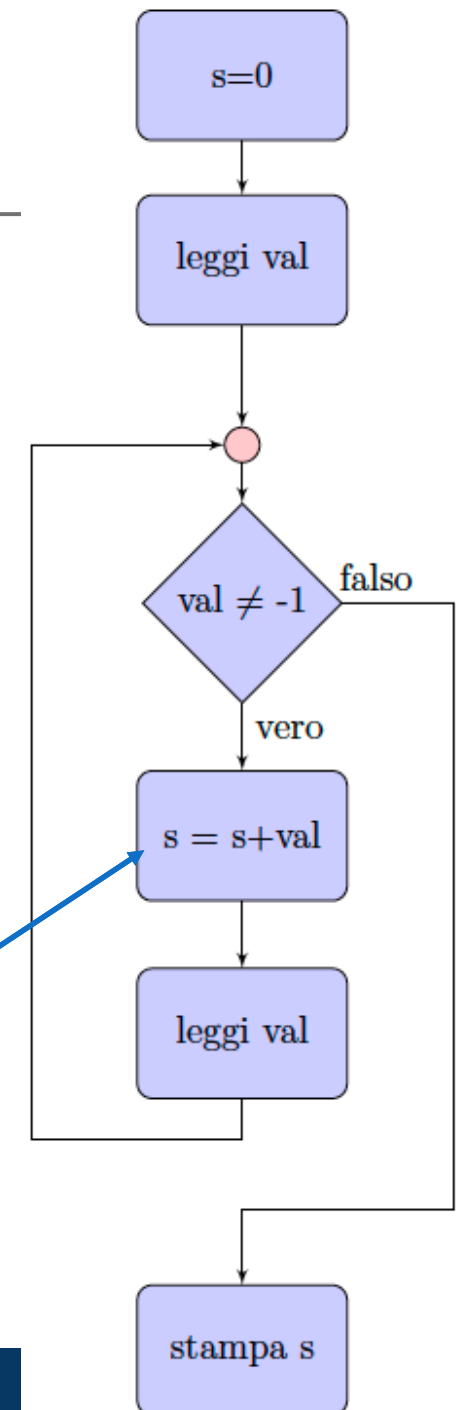


# Ciclo **while**

## *Esercizio:*

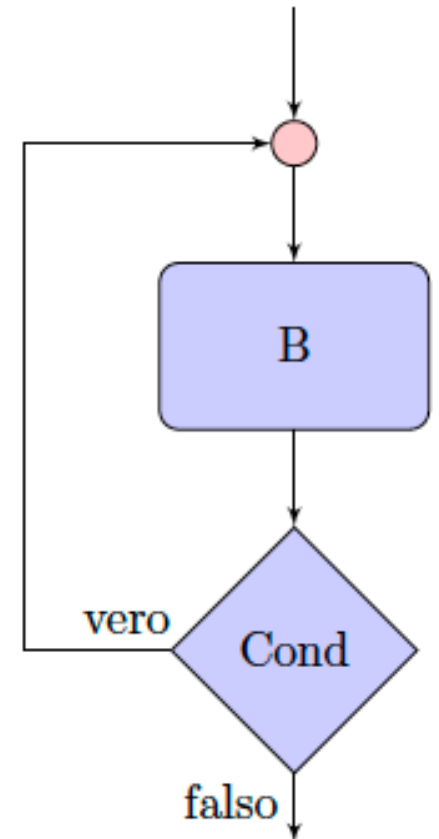
- Si vuole calcolare la somma di una serie di numeri naturali inseriti dall'utente da tastiera.
- La serie ha una lunghezza non nota;
- la fine della serie è riconoscibile dal valore -1 (detto tappo).
- Si noti che tale valore è riconoscibile come terminazione in quanto gli elementi della serie (per definizione di numero naturale) sono maggiori di zero.

accumulatore



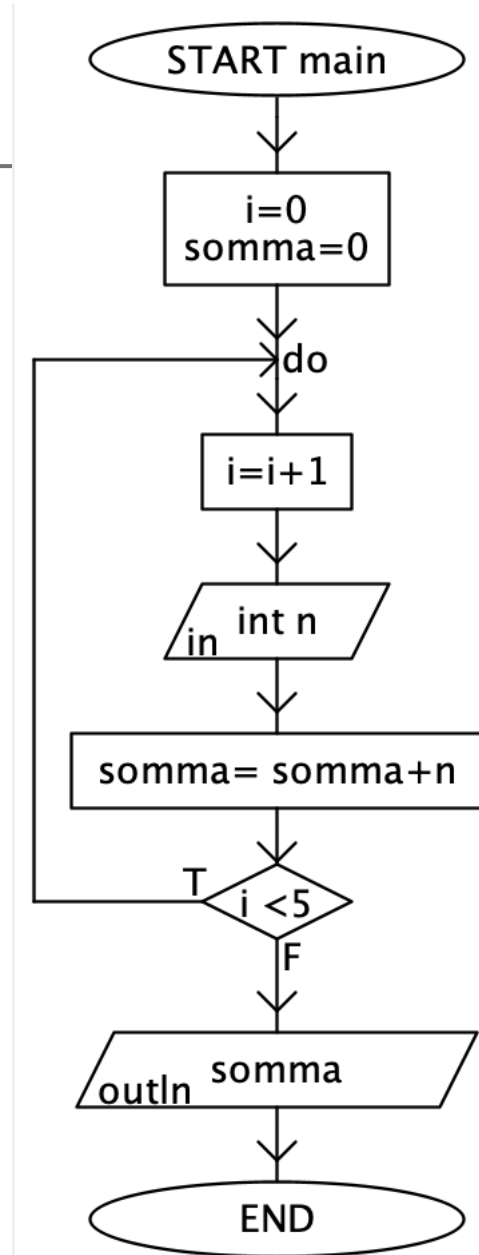
# Ciclo **do-while**

- Se si inverte la condizione del repeat-until:  
    *«Se Cond è vera il ciclo continua, altrimenti termina»*,
- si ottiene il ciclo **do-while**:
- *L'iterazione inizia con l'esecuzione del blocco B.*
- *Poi viene verificata la condizione Cond:*
- *se questa è vera ? si continua con la successiva iterazione,*
- *Altrimenti ? si esce dal ciclo;*
- *Cond rappresenta quindi la **condizione di continuazione del ciclo**.*



# Esempio

- Cosa rappresenta questo diagramma di flusso ?
- Che tipo di costrutto ciclico è usato?

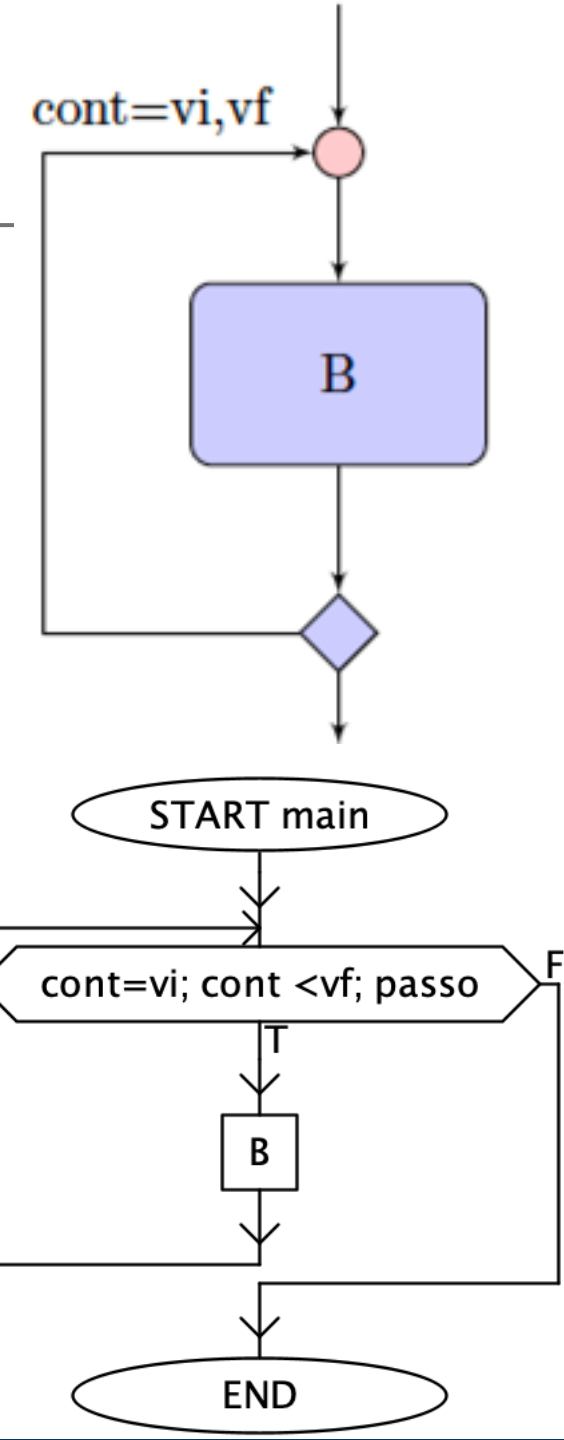




# Ciclo **for**

- Il ciclo for si usa in genere quando la condizione di permanenza in un ciclo è **predeterminata**.
- Tale ciclo consente di specificare quante volte debbano essere ripetuti l'istruzione o il blocco controllati dal ciclo.
- Si impiega una variabile contatore (di tipo intero) che conta il numero di iterazioni, o ripetizioni del ciclo
- Il ciclo for può esser parafrasato con "ripeti il blocco di istruzioni B per i valori del contatore cont che vanno da un certo valore iniziale vi a un certo valore finale vf con un certo passo" (solitamente 1).

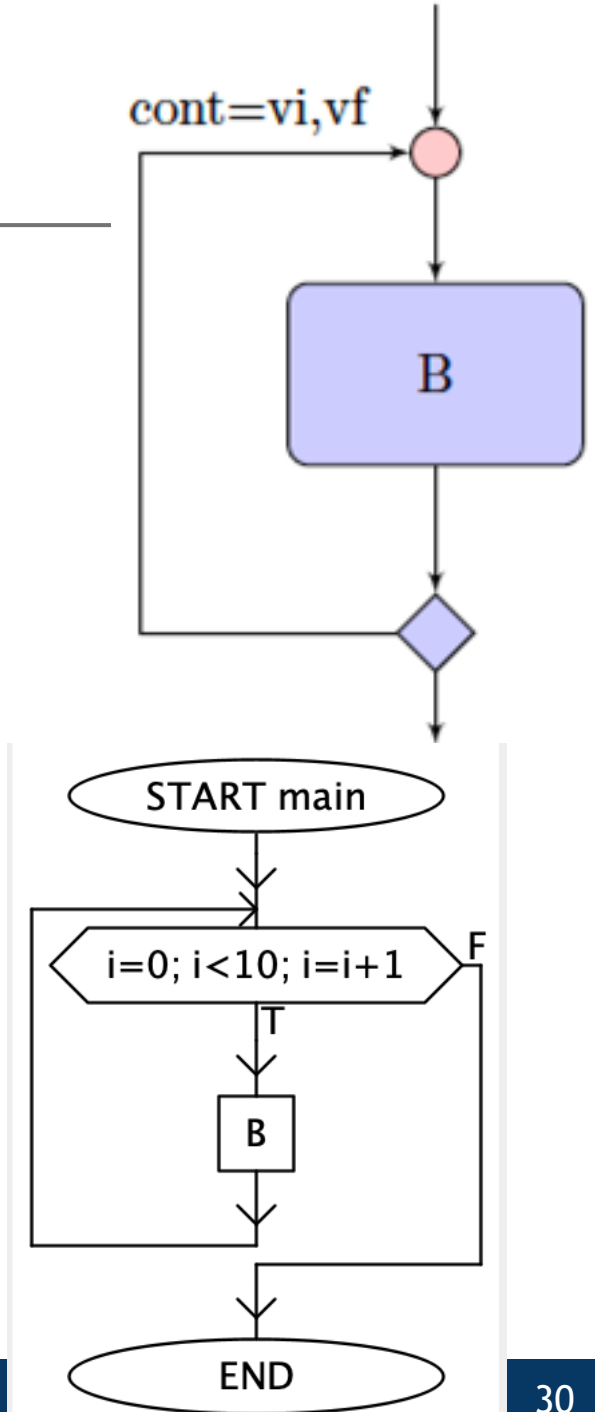
Notazione in  
Algobuild:



# Ciclo **for**

- Il ciclo for si usa in genere quando la condizione di permanenza in un ciclo è **predeterminata**.
- Tale ciclo consente di specificare quante volte debbano essere ripetuti l'istruzione o il blocco controllati dal ciclo.
- Si impiega una variabile contatore (di tipo intero) che conta il numero di iterazioni, o ripetizioni del ciclo
- Il ciclo for può esser parafrasato con "ripeti il blocco di istruzioni B per i valori del contatore cont che vanno da un certo valore iniziale vi a un certo valore finale vf con un certo passo" (solitamente 1).

Notazione in  
Algobuild:  
un esempio



# Esempio

- *Si realizzi un programma che legge il valore di un numero naturale  $n$ , e calcola la somma di tutti i numeri naturali compresi tra 1 e  $n$ .*

