

I FLOW CHART

Introduzione alla programmazione

1



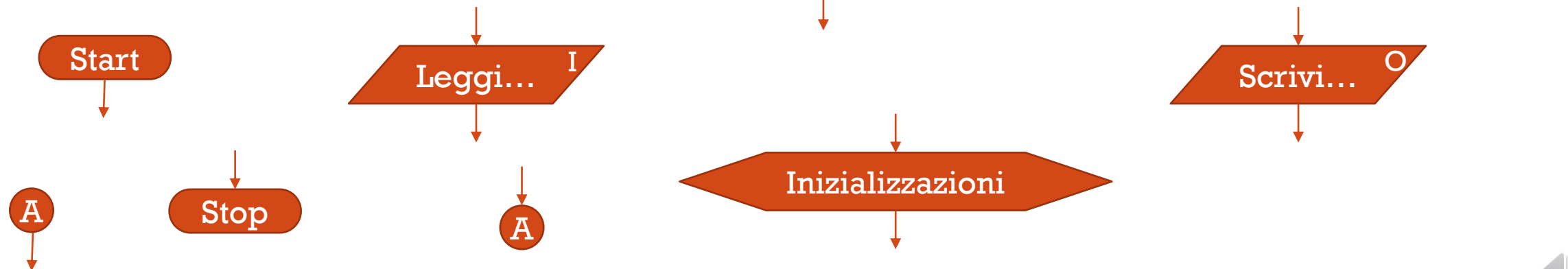
I FLOW CHART

- I flow chart permettono di rappresentare un algoritmo attraverso un diagramma a blocchi.
- Sono più efficaci e meno ambigui degli pseudo-linguaggi.
- Sono formati da un certo numero di blocchi predefiniti collegati da archi orientati.



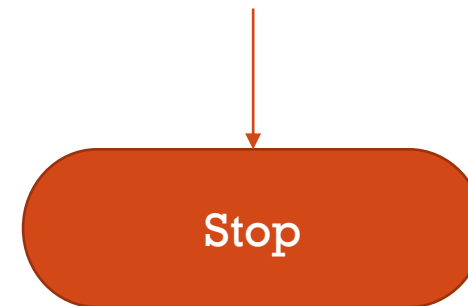
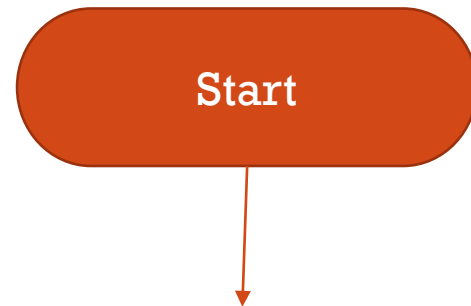
BLOCCHI PRINCIPALI

- I principali blocchi utilizzati in un flow-chart sono i seguenti:
 - blocchi di inizio/fine (start/stop);
 - blocchi di inizializzazione;
 - blocchi di lettura/scrittura (input/output);
 - blocchi di azione;
 - blocchi di controllo;
 - blocchi di richiamo.



BLOCCHI DI INIZIO/FINE (START/STOP)

- Indicano l'inizio e la fine del flow-chart (dell'algoritmo).
- Devono esserci un solo blocco di inizio ed uno di fine in ogni flow-chart.
- Dal blocco di «Inizio» (Start) parte un solo arco orientato (freccia).
- Al blocco di «Fine» (Stop) arriva un solo arco orientato (freccia).



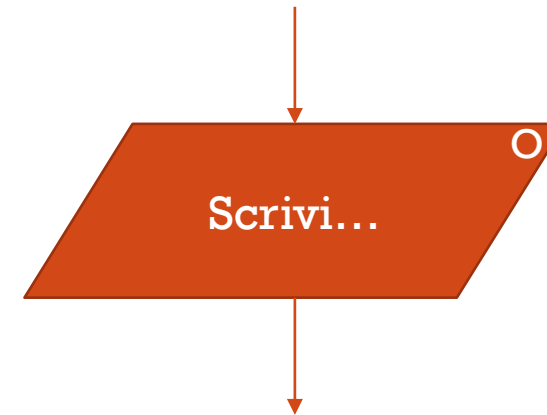
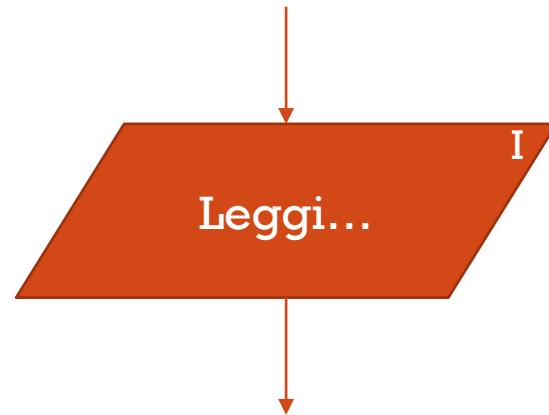
BLOCCHI DI INIZIALIZZAZIONE

- Indica il punto, normalmente è il secondo blocco del flow-chart ed è unico, in cui vengono definite ed inizializzate tutte le variabili e le costanti utilizzate nell'algoritmo.
- Hanno un solo arco orientato entrante ed uno uscente.



BLOCCHI DI LETTURA/SCRITTURA (I/O)

- Indicano i punti in cui sono previste operazioni di input/output dei dati.
- Hanno un solo arco orientato entrante ed uno solo uscente.
- A volte si inserisce una «I» o una «O» in alto a sinistra o in alto a destra nel blocco per distinguere i blocchi di input da quelli di output.



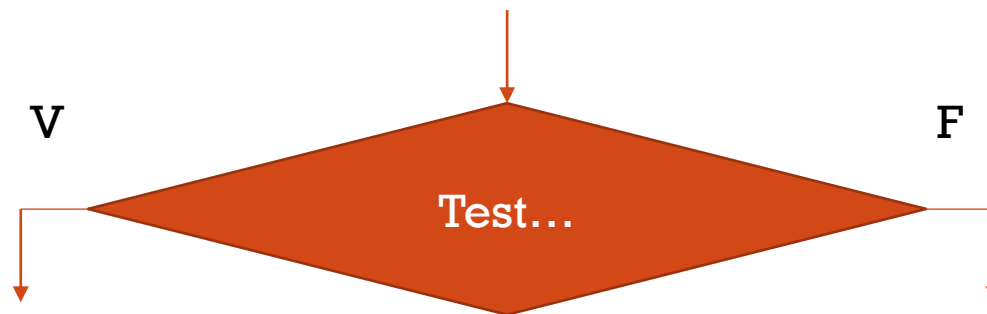
BLOCCHI DI AZIONE

- Indicano i punti vengono eseguite le istruzioni relative alle singole operazioni da compiere (somme, moltiplicazioni, assegnazioni, ...).
- Hanno un solo arco orientato entrante ed uno solo uscente.
- Sono, normalmente, i blocchi più numerosi all'interno del flow-chart.



BLOCCHI DI CONTROLLO

- Indicano i punti in cui vengono eseguite delle scelte in funzione del risultato di un test (condizione) binario.
- Il test è un'espressione composta da una o più espressioni di confronto ($>$, \geq , $<$, \leq , $=$, \neq , ...) collegate, attraverso gli operatori logici (and, or e not), in modo da formare un'unica proposizione logica che può portare solo a due risposte: Vero o Falso.
- Hanno un solo arco orientato entrante e due uscenti, uno per il «Vero» e uno per il «Falso».



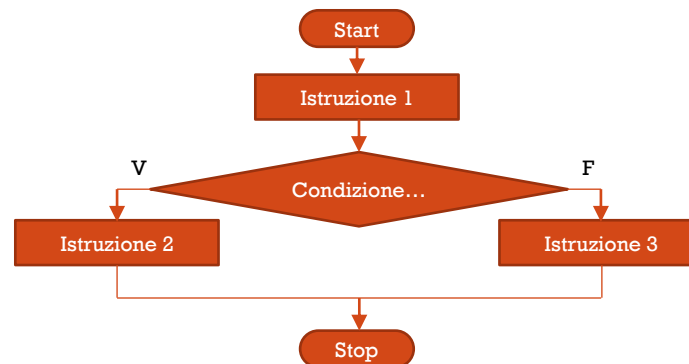
BLOCCHI DI RICHIAMO

- Permettono di dividere il diagramma in più parti.
- Hanno un solo arco che può essere o entrante o uscente.
- Il loro contenuto è un «tag», normalmente una singola lettera, che collega i blocchi tra loro.
- Il flusso entrante in un blocco con un determinato «tag» esce dal corrispondente blocco con il medesimo «tag».



PROGRAMMAZIONE STRUTTURATA

- Afferma che si può sviluppare un algoritmo utilizzando solo tre tipologie di costrutti:
 - la sequenza;
 - la selezione (o costrutto condizionale);
 - l'iterazione (o ciclo)
- Tutti questi costrutti, e l'algoritmo stesso, hanno una caratteristica in comune: «hanno un solo punto di ingresso ed un solo punto di uscita»



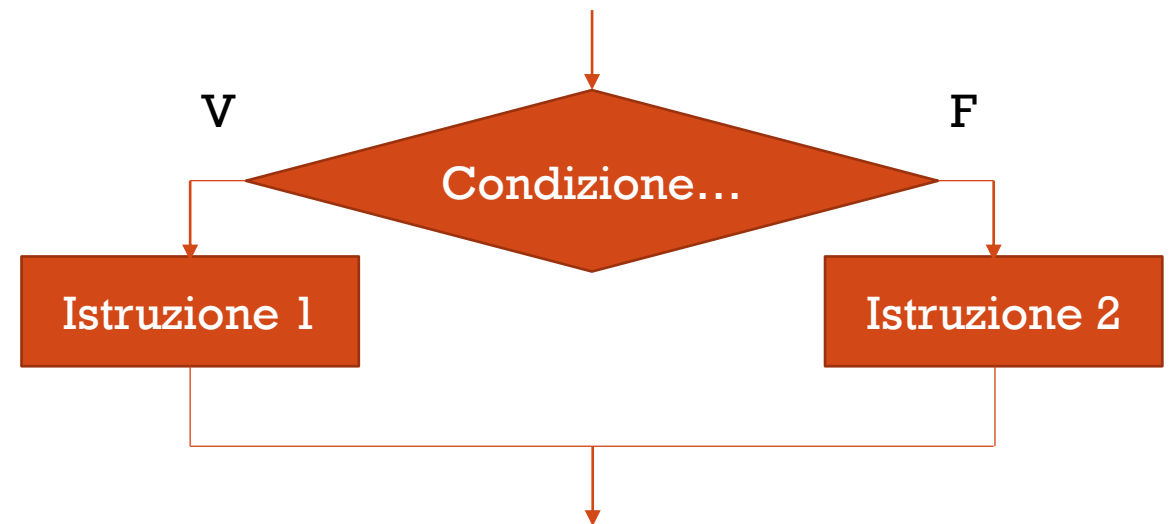
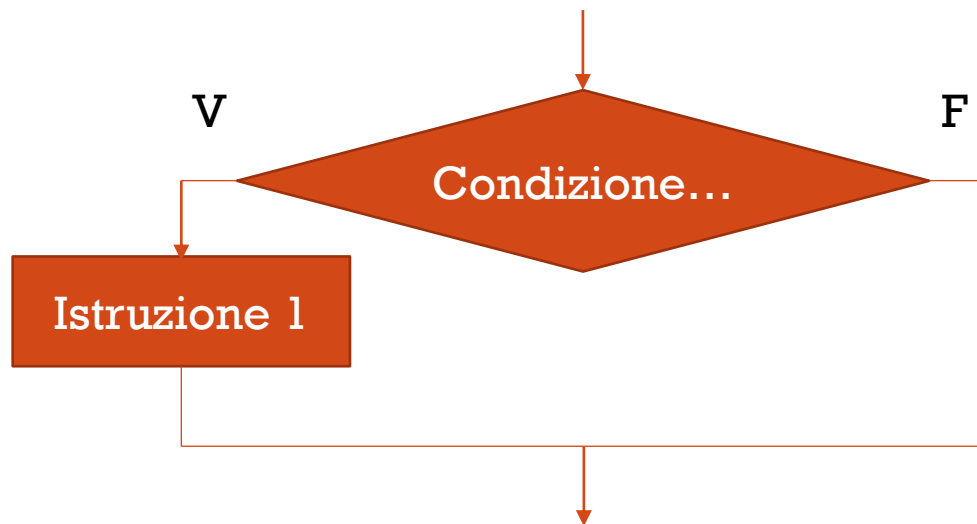
LA SEQUENZA

- Elenca le istruzioni nell'ordine in cui devono essere eseguite.



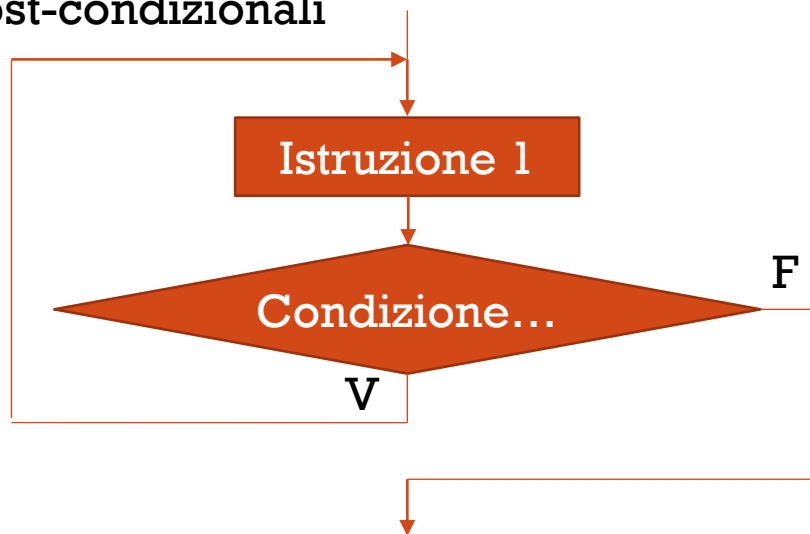
LA SELEZIONE

- Permette di scegliere fra due percorsi da seguire.
- La scelta dipende dalla valutazione di una condizione, che può essere Vera o Falsa.

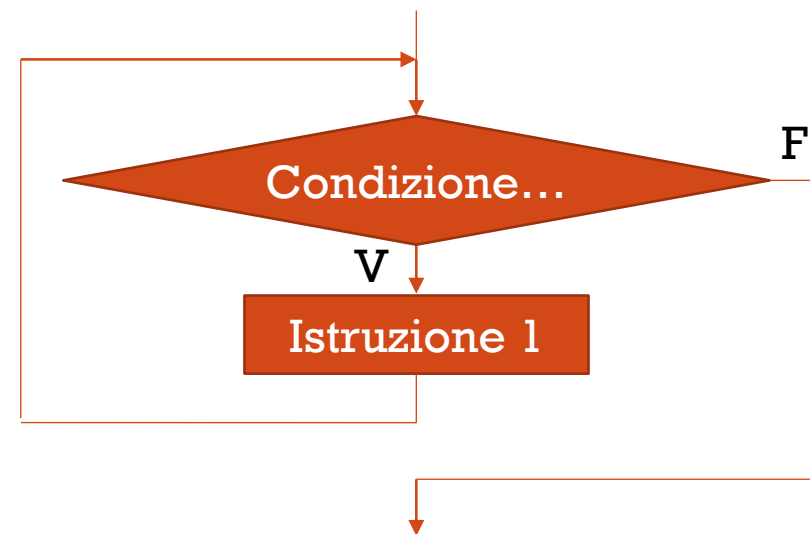


L'ITERAZIONE

- E' costituita da un blocco di istruzioni che vengono eseguite ripetutamente fino a quando una certa condizione cambia passando da Vero a Falso (o viceversa).
- Esistono due tipologie di costrutti iterativi:
 - pre-condizionali
 - post-condizionali



Post-condizionale

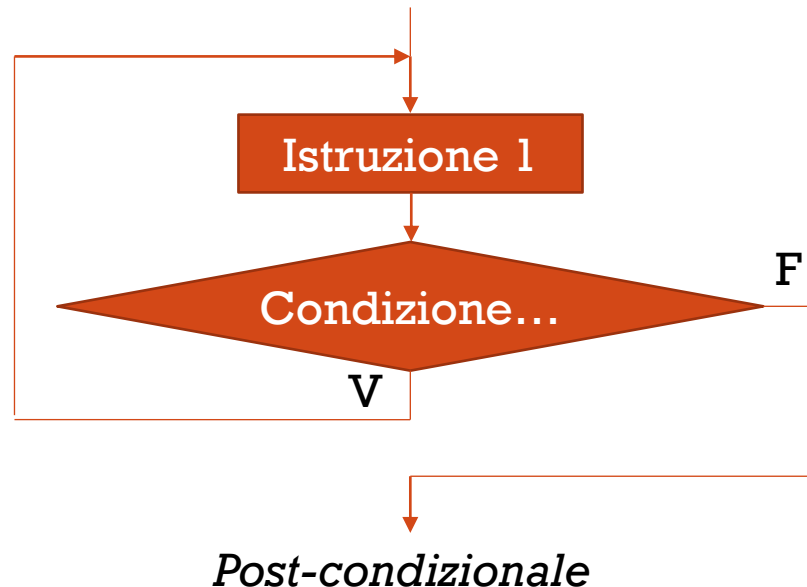


Pre-condizionale



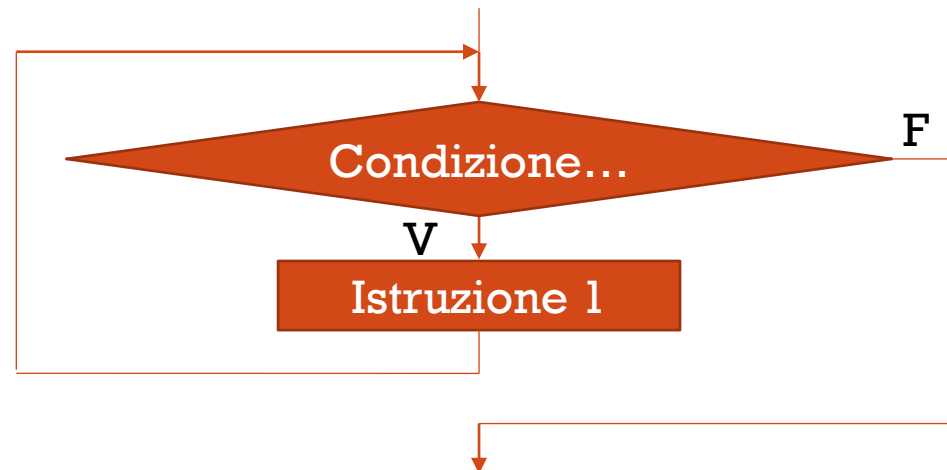
CICLO POST-CONDIZIONALE

- Prima sono eseguite le istruzioni che formano il blocco dell'iterazione e dopo viene valutata la condizione per stabilire se ripetere il ciclo oppure no.
- Viene utilizzato quando il corpo dell'iterazione deve essere eseguito almeno una volta.



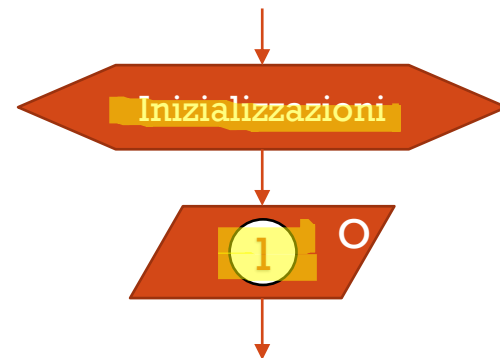
CICLO PRE-CONDIZIONALE

- Prima viene valutata la condizione e, poi, se è vera (o falsa), verranno eseguite le istruzioni che formano il blocco dell'iterazione.
- Se è stato eseguito il blocco delle istruzioni, al termine di questo si torna a valutare la condizione.
- Il blocco di istruzioni contenute nel ciclo potrebbe non essere mai eseguito.



LA LEGENDA

- Consente di avere un flow chart ordinato.
- Evita di dover scrivere molto testo all'interno dei blocchi.
- All'interno dei blocchi vengono inseriti dei riferimenti agli elementi della legenda.
- È composta da due sezioni principali:
 - definizioni e inizializzazioni delle variabili;
 - output.



Legenda

Inizializzazioni

`int n1, n2, n3`
`float media = 0`

Output

(1) `"Inserire un numero: "`



LA LEGENDA – CONVENZIONI

- Blocco «*Inizializzazioni*» :
 - per definire il tipo delle variabili si utilizzeranno i tipi fondamentali del linguaggio c:
 - `int` => numeri interi positivi e negativi;
 - `unsigned int` => numeri interi non negativi (≥ 0);
 - `float` => numeri in virgola mobile;
 - `char` => caratteri;
 - per indicare il valore a cui inizializzare una variabile si utilizzerà l'operatore "=":
 - `int var = 0`
 - per definire più variabili dello stesso tipo nella stessa riga, si separeranno le stesse attraverso la virgola:
 - `int var_1, var_2 = 1`



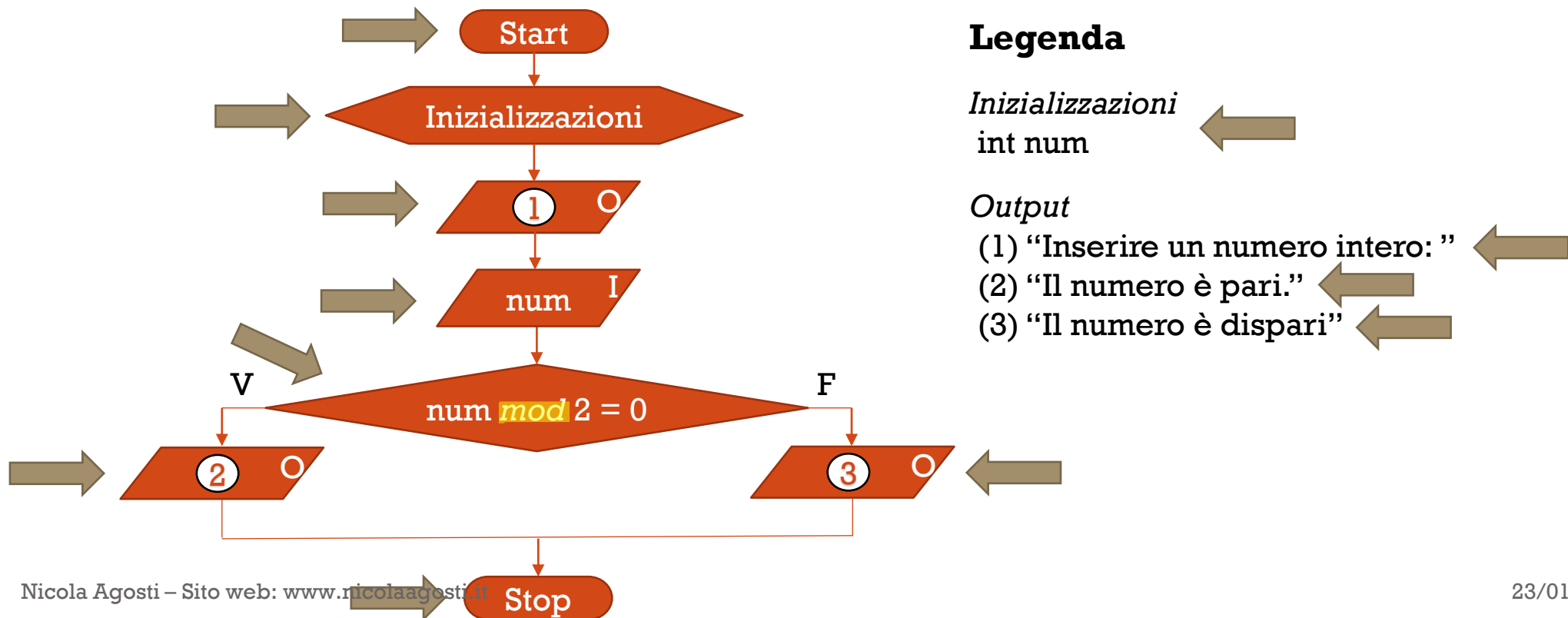
LA LEGENDA – CONVENZIONI

- Blocco «*Output*» :
 - per visualizzare una stringa costante si farà precedere la stringa dal segnaposto del corrispondente al blocco di output e poi si scriverà la stringa tra apici doppi:
 - (1) *“il mio primo output”*
 - per visualizzare il contenuto di una variabile si agirà come sopra e si indicherà il contenuto della variabile utilizzando il nome della stessa racchiuso tra le parentesi angolari:
 - (2) *“il contenuto della variabile è <var>”*



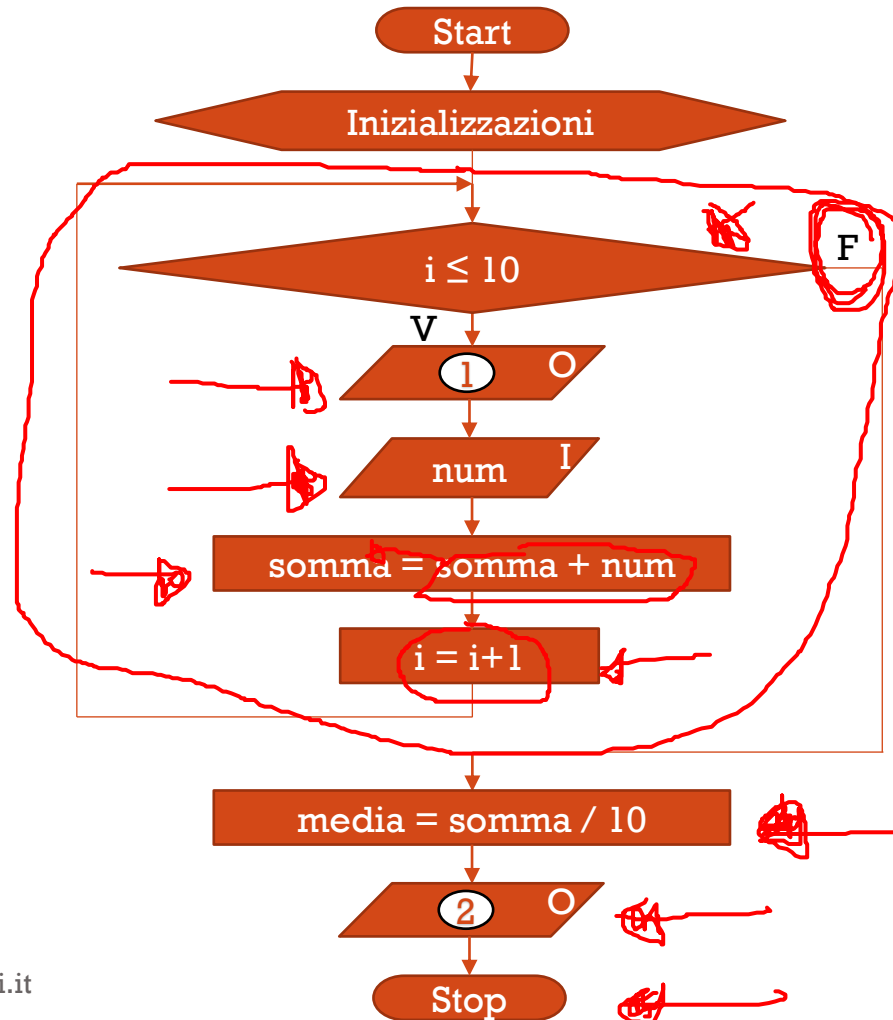
ESEMPIO: PARI/DISPARI

- Sviluppa il flow chart di un algoritmo che permetta di risolvere il seguente problema: “digitare sulla tastiera un numero intero e verificare se è pari o dispari”.



ESEMPIO: MEDIA TRA 10 NUMERI

- Sviluppa il flow chart di un algoritmo che permetta di risolvere il seguente problema: “calcolare e visualizzare la media dei 10 numeri inseriti dall’utente”.



Legenda

Inizializzazioni

`int i = 1`

`float num, somma = 0, media`

Output

- (1) “Inserire un numero (<i> di 10):”
- (2) “La media è: <media>”



ESERCIZI: SELEZIONE

- **Esercizio 1:** Dati due numeri interi positivi N_1 e N_2 , calcolare e visualizzare il numero massimo MAX . Descrivere il problema mediante flow chart.
- **Esercizio 2:** Dati tre numeri interi positivi N_1, N_2 e N_3 , calcolare e visualizzare il numero minimo MIN . Descrivere il problema mediante flow chart.
- **Esercizio 3:** Dato un numero intero positivo N , verificare se appartiene all'intervallo $[4, 10]$. Descrivere il problema mediante flow chart.
- **Esercizio 4:** Dato un numero intero positivo N , verificare se appartiene all'intervallo $[4, 10]$ o all'intervallo $[15, 20]$. Descrivere il problema mediante flow chart.

ESERCIZI: ITERAZIONE

- **Esercizio 1:** Dato N un numero intero positivo, generare e visualizzare in ordine crescente i primi N numeri interi positivi. Descrivere il problema mediante flow chart.
- **Esercizio 2:** Dato N un numero intero positivo, generare e visualizzare in ordine crescente i numeri dispari minori o uguali a N . Descrivere il problema mediante flow chart.
- **Esercizio 3:** Dato N un numero intero positivo, calcolare e visualizzare la somma dei primi N numeri pari. Descrivere il problema mediante flow chart.
- **Esercizio 4:** Dato un numero intero positivo N , trovare e visualizzare tutti i suoi divisori. Descrivere il problema mediante flow chart.

ESERCIZI: VARI

- **Esercizio 1:** Date due misure di tempo espresse in ore e minuti primi, calcolare la differenza fra la prima e la seconda misura esprimendola in ore e minuti primi. Descrivere il problema mediante flow chart.
- **Esercizio 2:** Dato un numero intero positivo N minore di mille, trasformarlo nella numerazione romana. Descrivere il problema mediante flow chart.
- **Esercizio 3:** Progettare, mediante flow chart, un algoritmo che effettui la lettura da tastiera di una serie di coppie di valori numerici. L'algoritmo deve calcolare e stampare il rapporto tra il valore minore e quello maggiore dei due. Il programma termina quando uno dei due valori o entrambi sono uguali a zero.
- **Esercizio 4:** Progettare un algoritmo che effettui le seguenti operazioni:
 - continui a leggere da tastiera una serie di terne di valori interi A , B e C finché non vengono inseriti dei valori tali per cui $A + B < C$;
 - conteggi il numero di volte in cui la differenza tra A e B è pari, dispari, e quando è nulla;
 - prima di terminare, visualizzi il valore dei valori conteggiati