



TANKERINO.COM

## Controllo del Flusso e Evoluzione delle Variabili in un Flowchart

Quando analizziamo un algoritmo tramite un flowchart, è fondamentale osservare come le variabili all'interno del sistema cambiano durante l'esecuzione.

Questo ci permette di comprendere non solo la sequenza logica, ma anche l'evoluzione delle variabili nel tempo e le decisioni prese dall'algoritmo basate su queste variabili.

Vediamo un esempio concreto: Chiedi all'utente un numero. Somma tutti i numeri fino al numero inserito. Infine stampa il risultato.

### Analisi:

#### Input:

- n: il numero fino al quale vogliamo sommare (con  $n > 0$ )

#### Variabili:

- somma: accumula la somma dei numeri
- i: contatore per il ciclo = 1

#### Output:

- somma

Ora creiamo la nostra tabella (trace table) che tiene conto di tutte le modifiche sulle variabili man mano che il nostro algoritmo procede con l'esecuzione.

---

Step	n	somma	i
Inizializzazione	Valore inserito	0	1

1	20	0	0
2	20	1	0

## Passi dell'Algoritmo:

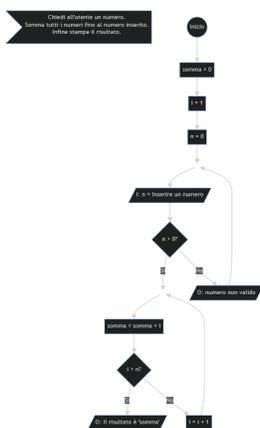
- 1) Inizializzazione delle Variabili: Qui definiamo il valore di partenza delle nostre variabili.
- 2) Chiediamo il dato di input all'utente
- 3) Ciclo di Somma: In questo punto, il programma inizia a sommare i numeri da 1 fino a "n"
- 4) Una volta contato fino ad n mostra la somma finale.

Ora, se visualizziamo la tabella e il flowchart contemporaneamente, possiamo vedere come le variabili cambiano nel tempo. Dopo ogni iterazione del ciclo, "somma" e "i" verranno aggiornati finché non raggiungono la condizione di uscita.

Il controllo del flusso è essenziale per comprendere l'evoluzione delle variabili e la logica dell'algoritmo. Il flowchart fornisce una rappresentazione visiva di come l'algoritmo opera, permettendoci di osservare le decisioni prese basate sulle variabili.

## Flowchart

Controlla con la trace table se è corretto il grafico!



-----

## Codice Mermaid

graph TB

id1>Chiedi all'utente un numero.

Somma tutti i numeri fino al numero inserito.

Infine stampa il risultato.]

start((Inizio)) -->

init\_somma[somma = 0 ] -->

init\_i[i = 1 ] -->

init\_n[n = 0 ] -->

hidden\_node\_1[ ] -->

input\_n[/I: n = Inserire un numero/] -->

check\_n{n > 0?}

check\_n -->|Si| hidden\_node\_2[ ]

check\_n -->|No| show\_message\_error[/O: numero non valido/]

show\_message\_error --> hidden\_node\_1

somma[somma = somma + i]

hidden\_node\_2 --> somma

somma --> check\_i

check\_i{i > n?}

check\_i -->|Si| print\_result[/O: Il risultato è 'somma'/]

check\_i -->|No| increment\_i[i = i + 1]

increment\_i --> hidden\_node\_2

style hidden\_node\_1 fill-opacity:0, stroke-opacity:0;

style hidden\_node\_2 fill-opacity:0, stroke-opacity:0;

-----  
  
(CC BY-NC-SA 3.0) lezione - by tankerino.com

<https://www.tankerino.com>

---

Questa lezione e' stata realizzata grazie al contributo di:



Risorse per la scuola

<https://www.baobab.school>



Siti web a Varese

<https://www.francescobelloni.it>