

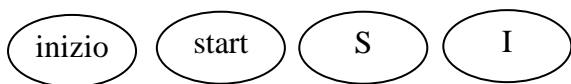
## COSA SONO I FLOW CHART

I flow chart sono schemi che descrivono visivamente come procede l'esecuzione di un programma. Essi non sono legati ad uno specifico linguaggio: dato un flow chart, il programmatore può poi usare un qualsiasi linguaggio di programmazione (si tratta, per così dire, di un linguaggio visuale comprensibile a tutti i programmatori). Il flow chart aiuta anche il programmatore a descrivere correttamente un algoritmo (il procedimento risolutivo di un problema).

Ogni tipo di istruzione che si può inserire in un programma ha un suo simbolo ed ognuna delle tre strutture fondamentali della programmazione (sequenza, selezione ed iterazione) può essere rappresentata. Esistono anche simboli speciali (inizio programma, fine programma ecc.) che non rappresentano istruzioni vere e proprie ma che sono utili per la costruzione del flow chart.

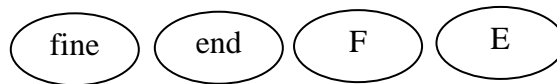
### I PRINCIPALI SIMBOLI

**Inizio programma** (i simboli sono equivalenti)  
puoi ricordare solo un simbolo a scelta ...



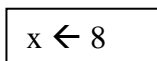
'S' sta per Start, 'i' sta per Inizio

**Fine programma** (i simboli sono equivalenti)  
puoi ricordare solo un simbolo a scelta ...

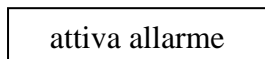


'F' sta per fine, 'E' sta per End

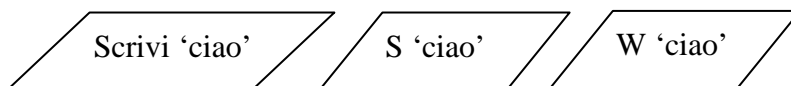
**Assegnamento o altre istruzioni generiche**  
(esempio: dai ad x il valore 8)



(esempio di istruzione generica)



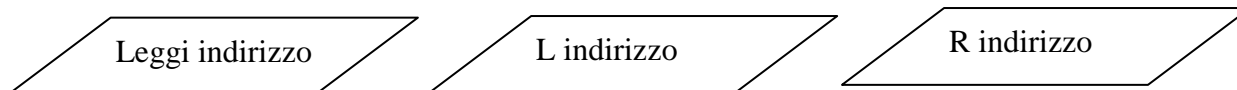
**Scrittura** di un valore sul video (i simboli sono equivalenti)  
puoi ricordare solo un simbolo a scelta ...



'S' sta per scrivi, 'W' sta per Write

**Lettura** di un valore scritto con la tastiera e memorizzato nella variabile indicata  
(i simboli che seguono sono equivalenti: ricordane uno a tua scelta ..)

Esempio: leggere dove abita una persona e memorizzare l'informazione nella variabile *indirizzo*



'L' sta per Leggi, 'R' sta per Read

## ALTRI ESEMPI

$$x \leftarrow y$$

dai ad x lo stesso valore di y

$$x \leftarrow y - 1$$

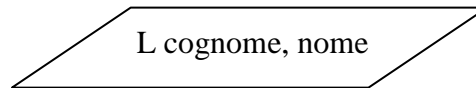
dai ad x il valore di y diminuito di 1

$$x \leftarrow (5 - 2) * (4 + 3)$$

dai ad x il valore dell'espressione  $(5 - 2) * (4 + 3)$



scrivi sul video prima il valore della variabile x e poi quello della variabile y



accetta dalla tastiera un primo valore che memorizzerai nella variabile 'cognome' e poi un secondo valore che memorizzerai nella variabile 'nome'

I simboli visti fino ad ora servono per istruzioni singole. Vediamo ora come si rappresentano le strutture fondamentali della programmazione (sequenza, selezione, iterazione).

### Sequenza

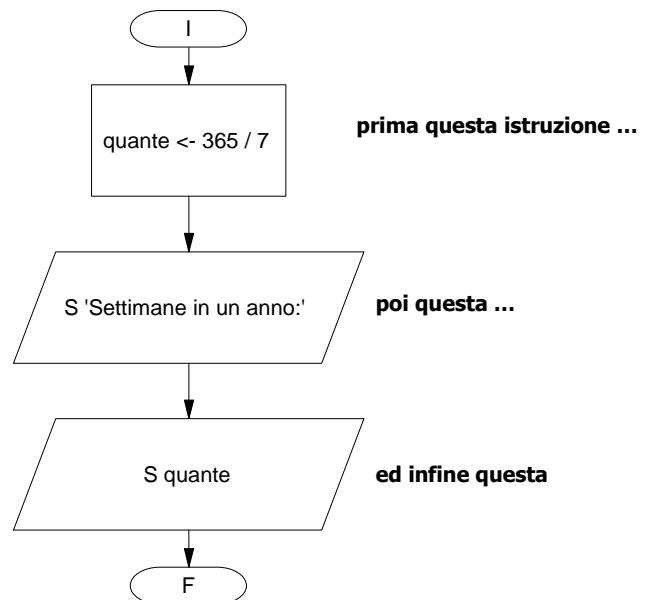
Per indicare che due istruzioni vanno eseguite una dopo l'altra si mettono i loro simboli uno sotto l'altro collegandoli con una freccia. Ecco un esempio.

*Calcoliamo quante settimane ci sono in un anno dividendo il numero dei giorni per 7.*

Memorizziamo prima il risultato della divisione nella variabile *quante*.

Poi scriviamo un messaggio sul video che 'annuncia' il risultato

Infine scriviamo il risultato, cioè il valore attuale della variabile *quante*.



Il senso della freccia ↓ indica il flusso di esecuzione.

## Selezione (decisione, scelta, alternativa)

Il solo fatto di poter indicare un'istruzione dopo l'altra (sequenza) non ci consentirebbe di sviluppare programmi interessanti. Ci sono innumerevoli situazioni in cui è necessario fare un 'controllo' ed agire di conseguenza.

### Esempi

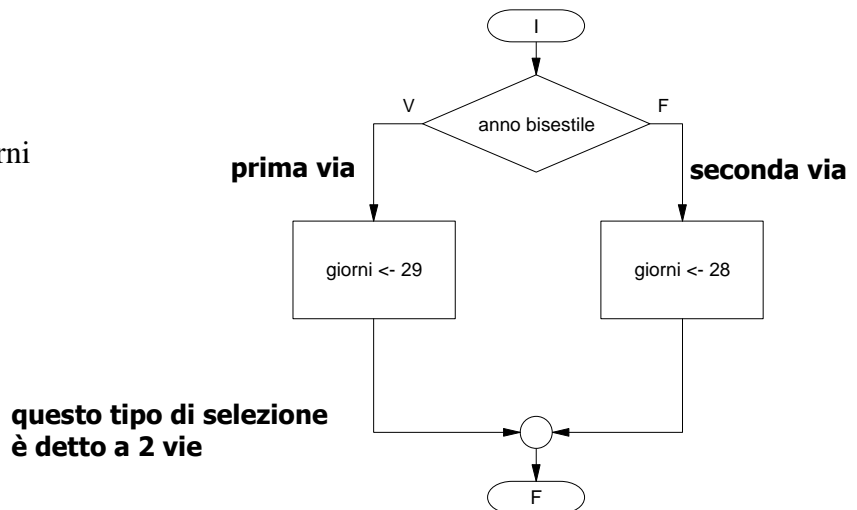
Se l'anno è bisestile allora considera febbraio con 29 giorni, altrimenti consideralo con 28.

Se l'età è minore di 18 allora applica sconto, altrimenti applica prezzo pieno.

Se la temperatura supera 37 C allora fai suonare l'allarme.

Come avrete osservato, si controlla una *condizione* (anno bisestile ?, età minore di 18 ?, temperatura supera 37 ?) che può risultare vera o falsa. In qualche caso (primi due esempi) ci sono operazioni (diverse) sia nel caso la condizione risulti vera sia nel caso risulti falsa. E' però possibile (terzo esempio) che nel caso la condizione sia falsa non ci sia nulla da fare. Ecco come si rappresenta in un flow chart questo modo di procedere:

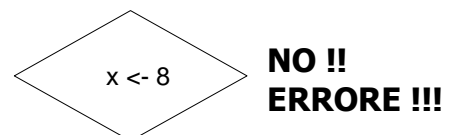
Se l'anno è bisestile allora  
considera febbraio con 29 giorni  
altrimenti  
consideralo con 28



La condizione deve essere scritta all'interno del rombo. Il flusso di esecuzione si divide a seconda del risultato del controllo (V sta per Vero e F sta per Falso) e solo la strada 'a destra' o 'a sinistra' viene percorsa.

Le istruzioni nella sezione 'vera' (o 'falsa') possono essere anche molte e non una sola come nel nostro esempio. Quando le istruzioni da eseguire a condizione vera (o falsa) sono terminate il flusso si ricongiunge usando il simbolo  $\bigcirc$  chiamato *connettore*.

Attenzione: nel rombo potete mettere solo condizioni, non assegnamenti! Sarebbe quindi un errore grave il seguente:

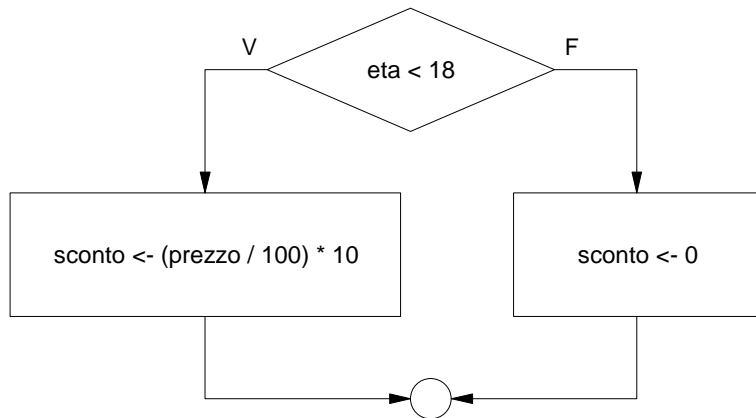


Se l'età è minore di 18 allora

applica sconto

altrimenti

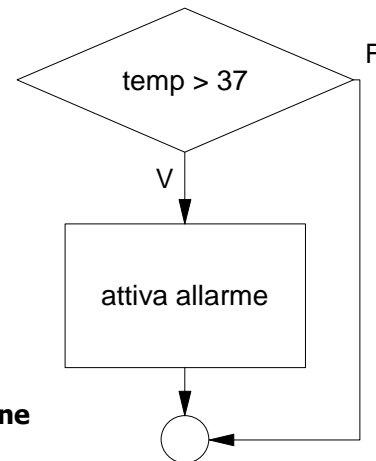
applica prezzo pieno.



Nota: nel flow chart immaginiamo che le variabili *eta* e *prezzo* abbiano già un valore valido e che i calcoli indicati siano quindi fattibili. Lo sconto (10%) viene calcolato dividendo il prezzo pieno per 100 (calcolando così l'1%) e moltiplicando il risultato per 10 (ottenendo il 10%).

Se la temperatura supera 37 C allora  
fai suonare l'allarme.

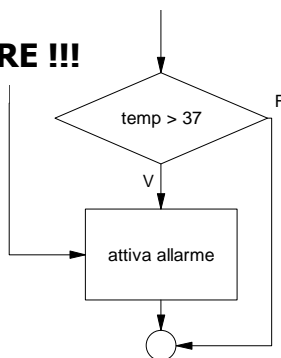
Come potete vedere se non c'è nulla che deve essere fatto quando la condizione è falsa, si congiunge la parte 'falso' direttamente con il connettore.



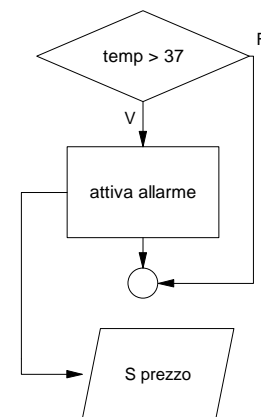
questo tipo di selezione è detto a una via

NOTA IMPORTANTE: in ogni diagramma che rappresenta la selezione ci deve essere sempre un solo punto di ingresso (la freccia entrante nel rombo) ed un solo punto di uscita (il connettore finale). Evitate quindi frecce che dall'interno escono verso altri punti del diagramma o che dall'esterno giungono in un punto interno:

**NO !!  
ERRORE !!!**

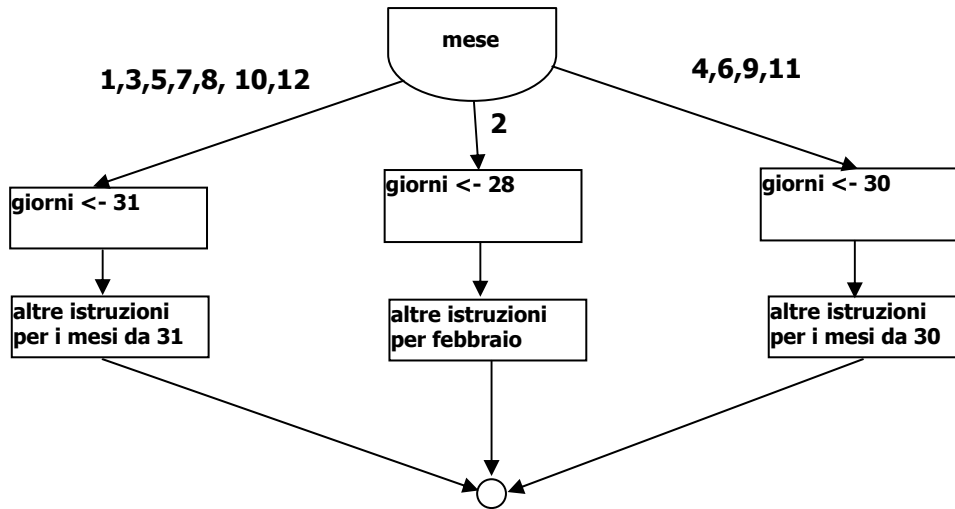


**NO !!  
ERRORE !!!**



### Selezione a molte vie

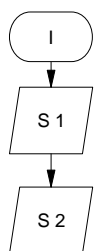
Esiste anche un'utile variante che prevede molte vie a seconda del valore di una variabile. Immaginiamo che la variabile *mese* contenga un valore da 1 a 12 che rappresenta uno dei mesi dell'anno. Di nuovo, vorremmo sapere quanti giorni considerare. Vi ricordo che i mesi con 31 giorni sono 1 (gennaio), 3 (marzo), 5 (ecc.), 7, 8, 10, 12; quelli con 30 giorni sono 4 (aprile), 6, 9, 11; immaginiamo per semplicità che febbraio (2) ne abbia sempre 28.



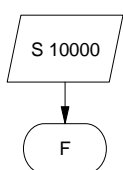
Il flusso può seguire tre vie a seconda del valore della variabile *mese*. I tre flussi possibili si ricongiungono poi nel solito connettore.

### Iterazione (ripetizione, cicli)

Pur avendo a disposizione sequenza e iterazione rimangono molte le situazioni 'intrattabili' con questi strumenti. Consideriamo infatti questo problema all'apparenza molto semplice: stampare i numeri da 1 a 10000. Certamente la soluzione di usare per 10000 volte l'istruzione di scrittura per stampare ogni numero non è molto praticabile ...

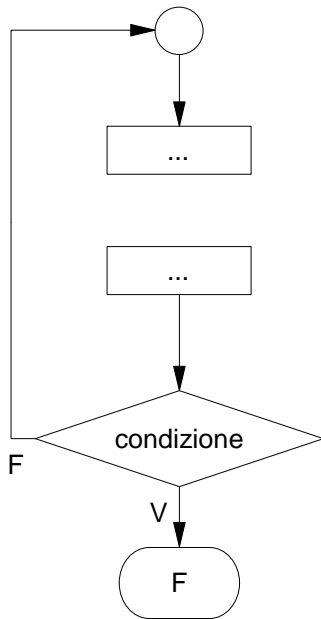


A parte i problemi di trovare uno spazio sufficiente per disegnare schemi così enormi ed il tempo per disegnarli, saremmo costretti a modificare sostanzialmente il diagramma al variare della richiesta (stampare solo i primi 5000 numeri o fino al 15000).



Partendo invece dalla considerazione che si stanno ripetendo istruzioni molto simili (cambia solo in valore da scrivere sul video) si perviene ad una struttura ciclica che fa ripetere una od un gruppo di istruzioni fino al raggiungimento di una condizione (in questo caso il raggiungimento del 10000).

Ecco lo schema generico (cioè non applicato alla soluzione di alcun problema) di un flow chart per la struttura iterativa.



Le istruzioni sono ripetute una prima volta. Giunti alla condizione, se questa è vera allora si esce ed il ciclo termina. Se la condizione è falsa si ritorna all'inizio (connettore di ingresso).

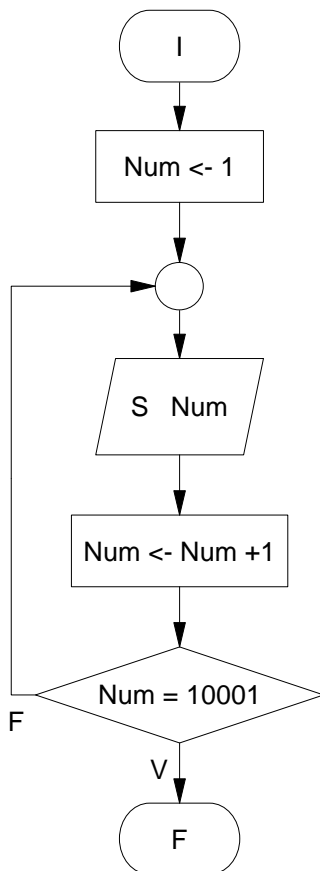
Se all'interno del ciclo, prima o poi, una delle istruzioni farà sì che la condizione diventerà vera, il ciclo terminerà veramente.

Diversamente procederà all'infinito (come potrebbe servire per il controllo di un processo industriale).

Questo tipo di ciclo è detto 'con ripetizione per falso', o a 'uscita per vero' ed ancora 'con controllo in coda' (cioè alla fine del ciclo).

Notiamo ancora: *con questo tipo di ciclo le istruzioni vengono eseguite almeno una volta.*

Vediamo il flow chart completo per la stampa dei numeri da 1 a 10000

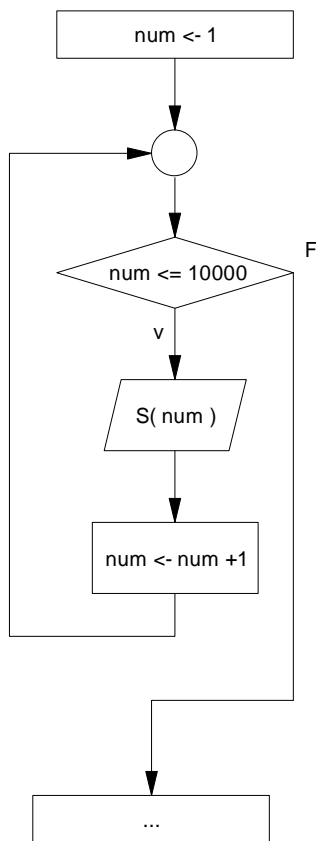


La variabile *Num* è usata per controllare l'uscita dal ciclo. Essa viene impostata al valore 1 *prima* di iniziare il ciclo.

Il ciclo inizia stampando sul video il valore della variabile *Num* (la prima volta stamperà quindi 1).

La variabile *Num* viene incrementata di 1.

Se la variabile *Num* ha raggiunto il valore 10001 (quindi ha già stampato il 10000) si esce, altrimenti si ritorna all'inizio (verrà stampato sul video un nuovo numero, si incrementa ancora il valore di *Num* e così via ...).



Ed ecco un esempio di flow chart per l'altro tipo fondamentale di ciclo, quello detto con controllo in testa (all'inizio) ed uscita per falso.

Infatti fintanto che la condizione rimane vera il ciclo viene ripetuto. Non appena la condizione diviene falsa il ciclo termina.

Il ciclo potrebbe anche non cominciare neppure, se la condizione fosse da subito falsa. Il ciclo visto in precedenza, invece, controllando la condizione alla fine, almeno una volta comunque esegue le istruzioni (e questo potrebbe essere indesiderabile).