Informatica

Capitolo 1 – Introduzione

Prof. Ivan Gentile

Informatica

- Contrazione di **Informazione** e **Automazione**
- Trattamento automatico delle informazioni
- In inglese: **Computer Science**
- *L'informazione* ha due accezioni
  - Dati che vanno conservati e/o elaborati;
  - probabilità di certi messaggi

Teoria dell'informazione

- Nata nel 1948
- *Claude Elwood Shannon* USA, 1916-2001) *Warren Weaver* (USA 1894-1978)
- Si occupa dei problemi della trasmissione dei messaggi
- Più opportuno **Teoria della comunicazione**.

Definizioni

- Un **algoritmo** è una sequenza finita di istruzioni non ambigue.
  - Parola relativamente recente (circa 1950)
  - Ha sostituito **algorismo** (calcolare con numeri arabi)
  - Deriva da **Al-Khwarizmi**, grande matematico arabo del IX secolo
- Un **programma** è un algoritmo scritto in uno specifico linguaggio detto appunto **linguaggio di programmazione**.
- Un **processo** è un programma in corso di esecuzione.

Esempio n.1: `(h, m, s) -> st`

- Vogliamo convertire un orario espresso in ore (`h`), minuti (`m`), secondi (`s`) in secondi totali (`st`)
- Esempio: `3h, 20m, 10s` diventa `3 * 3600 + 20 * 60 + 10 = 12010st`
- Algoritmo
  - Acquisire `h, m, s`
  - Prendere il numero di ore e moltiplicarlo per `3600`, questo nuovo valore lo chiamo `sh` (secondi in un'ora)
  - Prendere il numero di minuti e moltiplicarlo per `60`, questo nuovo valore lo chiamo `sm` (secondi in un minuto)
  - Sommare `sh, sm` e `s` questo numero lo chiamo `st`
  - Restituire il valore `st`

Es.1: proviamo ad "accorciare"

- Acquisire h, m, s
- sh = h * 3600
- sm = m * 60
- st = sh + sm + s
- Restituire il valore st

Nota:
h,m,s sono dati di INPUT
st è un dato di OUTPUT
3600 e 60 sono dati interni COSTANTI

Es.2 st -> (h,m,s)

- Vogliamo sapere un certo numero di secondi (st) quante ore (h), minuti (m), secondi (s) sono
- Esempio: `3900 st` corrispondono a `1h, 5m, 0s`
- Algoritmo
  - Acquisire `st`
  - `h = st /3600` (divisione intera)
  - `st2 = st - h * 3600` (secondi che restano dopo aver tolto le ore)
  - `m = st2/ 60 (divisione intera)`
  - `s = st2 – st2 * 60`
  - Restituire `h, s, t`

Nota:
st: Input
h,m,s: OUTPUT
3600 e 60 sono dati interni COSTANTI

Selezione

- Non sempre dobbiamo fare le stesse operazioni
- Ma in alcuni casi dobbiamo fare delle cose, in altri casi altro

Esempio: Flat-Tax

- Dato un reddito lordo RL calcolare il reddito netto RN e le tasse pagate T sapendo il valore dell'aliquota fiscale Al (l'aliquota è indipendente dal RL: flat-tax)
- Esempio: `RL = 32000€`, Al = 23%, in tal caso RN = 24640 €, T = 7360€
- Algoritmo
  - Acquisisci `RL, Al`
  - `T = RL * Al /100;`
  - `RN = RL – T;`
  - Restituisci T e N

Tassazione Variabile

- Supponiamo che la tassazione **dipenda dal reddito**
- Esempio abbiamo 3 aliquote (Al1, Al2, Al3) e due redditi oltre il quale cambia l'aliquota (R1, R2)
  - `0 < RL < R1` si applica l'aliquota `Al1`
  - `R1 <= RL < R2` si applica l'aliquota `Al2`
  - `RL >= R2` si applica `Al3`

Soluzione

- Con la sola "sequenza" non è possibile risolvere questo problema
- E' necessario **selezionare** l'aliquota da applicare in base al reddito
- Algoritmo
- Acquisisci RL, RN, T, R1, R2, Al1, Al2, Al3
- Se RL < R1 allora T = RL * Al1 /100
- Se invece RL <R2 allora T = RL * Al2 /100
- Se invece, RL >= R2 allora T = RL * Al3 /100
- In ogni caso RN = RL – T
- Restituisci RN, T

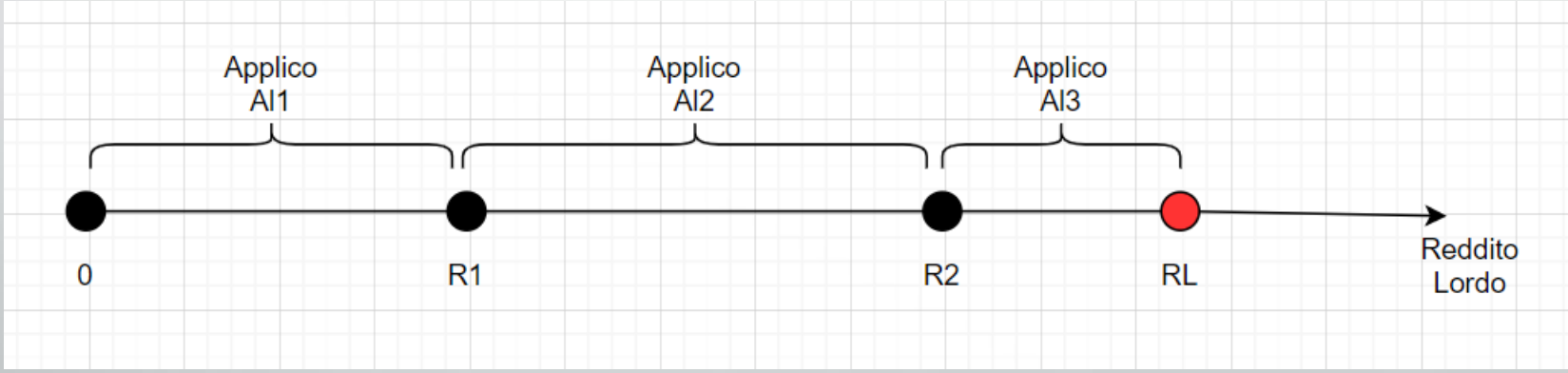
Nota: `R1, R2, Al1, Al3, Al3` sono dati che cambiano "raramente": meglio non chiederli in input ma come dati Interni

Questa tassazione variabile NON è equa

- Esempio (IRPEF 2024)
  - `Al1 = 23%, Al2 = 35%, Al = 43%`
  - `R1 = 28000 €, R2=50000€`
- Supponiamo che
  - Alice guadagna (reddito lordo) 28000€ mentre Bob guadagna 27999€
  - Reddito netto di Alice: `RN = RL – Al2/100 *RL= 28000 – 0.35* 28000 = 28000 – 98000 =` **`18200€`**
  - Reddito netto di Bob: `RN = RL – Al2/100 *RL= 27999 – 0.23* 28000 = 27999 – 6440 =` **`21559€`**
  - **Alice con un reddito lordo in più prende un netto minore (di 3359€!!!)**

Togliamo l'ingiustizia: Aliquota **Progressiva**

- Il reddito è diviso in varie parti
- Ad **ogni parte è applicata una tassazione diversa**



Esempio della tassazione progressiva

Esempio (IRPEF 2024)

`Al1 = 23%, Al2 = 35%, Al = 43%`

`R1 = 28000 €, R2=50000€`

Supponiamo che

Alice guadagna (reddito lordo) 28000€ mentre *Bob* guadagna 27999€

- Reddito netto di Alice: RN
  - `Prima Tassazione T1 = 0.23* 27999 = 6439.77`
  - `Seconda Tassazione T2 = 0.35 * 1 = 0.35`
  - **`Tassazione totale T = T1+T2 = 6439.77 + 0.35 = 6440.12 €`**
  - **`Reddito Netto RN = RL – T = 28000 – 6440.12 = 21559.88€`**

- Reddito netto di Bob: RN
  - `Prima Tassazione T1 = 0.23* 27999 = 6439.77`
  - **`Tassazione totale T = T1 = 6439.77 €`**
  - **`Reddito Netto RN = RL – T = 27999 – 6439.77 = 21559.23€`**

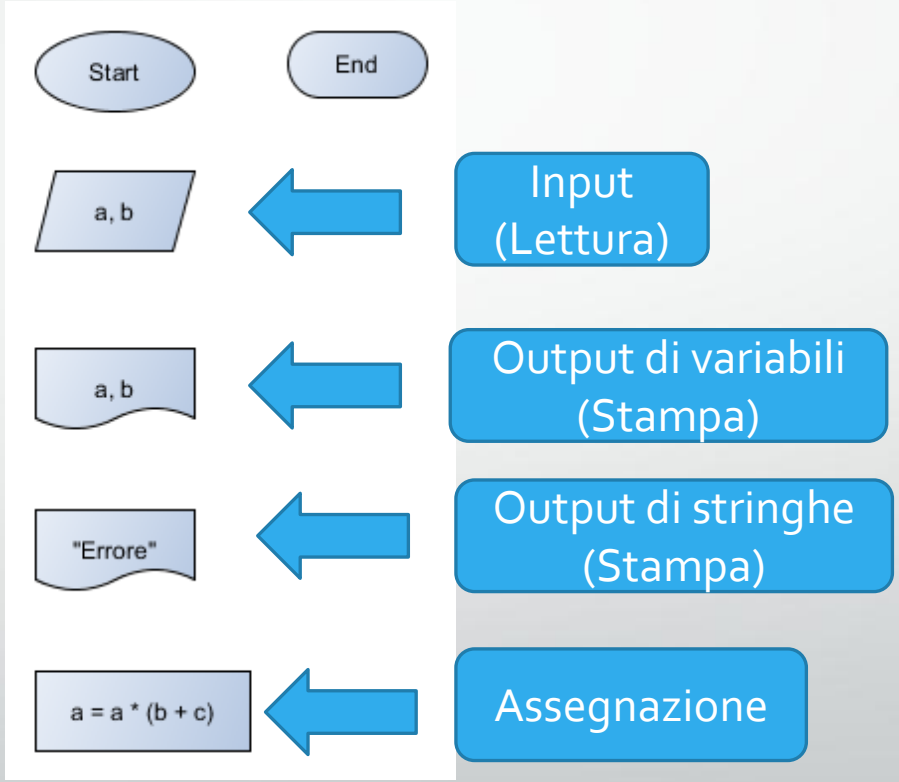
Alice con un reddito lordo maggiore prende SEMPRE un netto maggiore

Algoritmo aliquota progressiva

- E' più complesso da esprimere in linguaggio naturale
- Abbiamo la necessità di pensare a un linguaggio che ci **agevoli** la scrittura e la lettura degli algoritmi
- La prima cosa da pensare è un linguaggio Grafico: **Flow Chart**

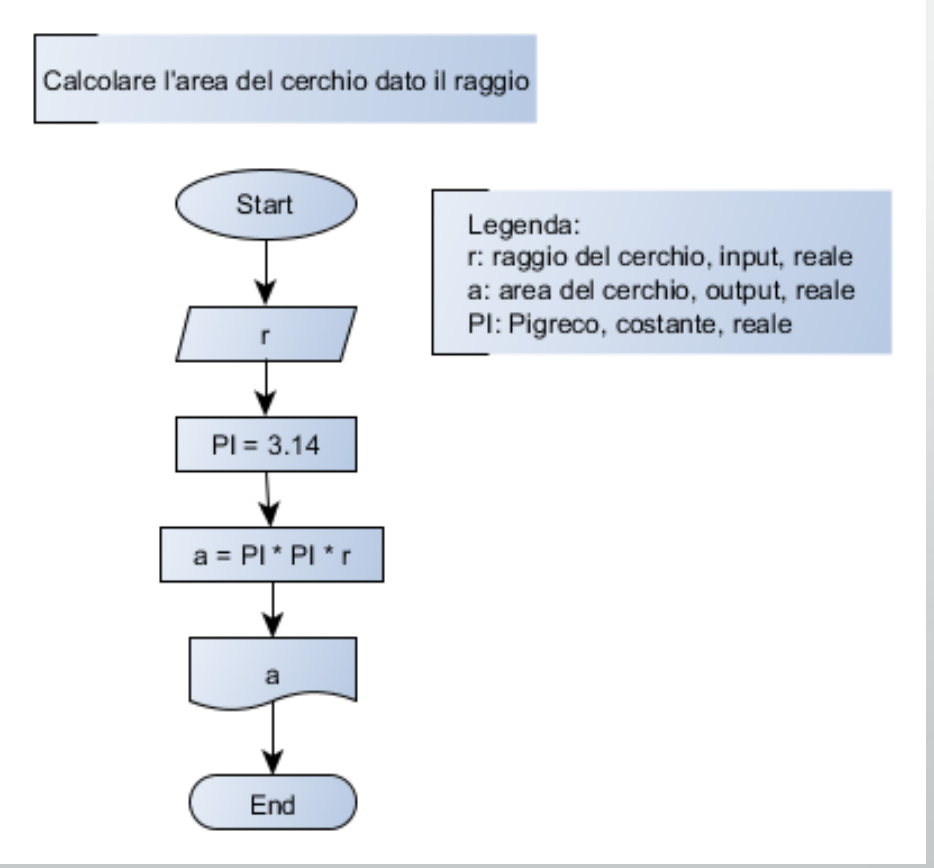
Flow-Chart (Diagramma di Flusso)

Rappresentazione grafica di un algoritmo

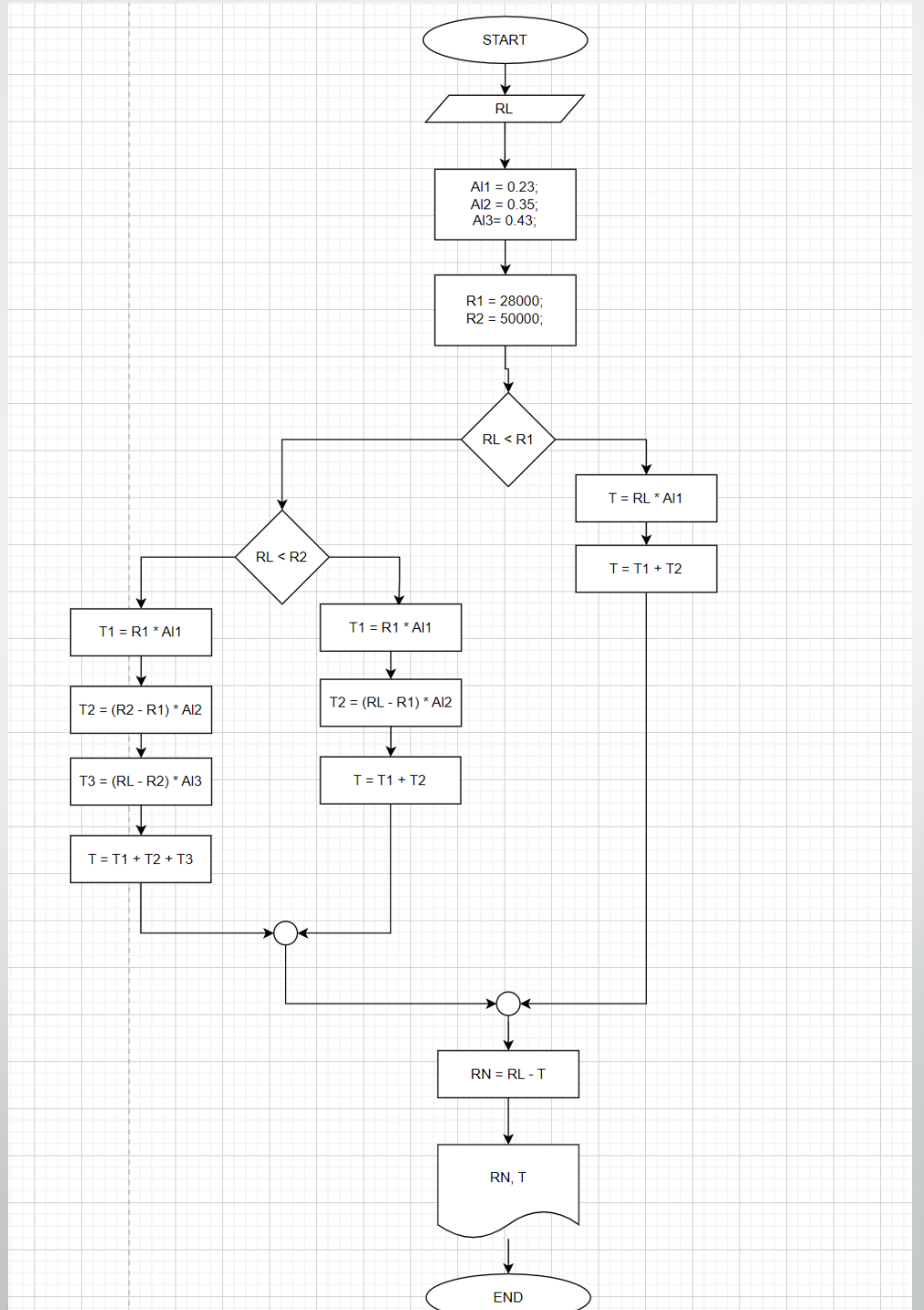


01 – AreaCerchio

- Testo: Calcolare e stampare l'area di un cerchio dato il raggio.
- Nome cartella: `01-AreaCerchio`
- https://youtu.be/LfCTY3JDs04



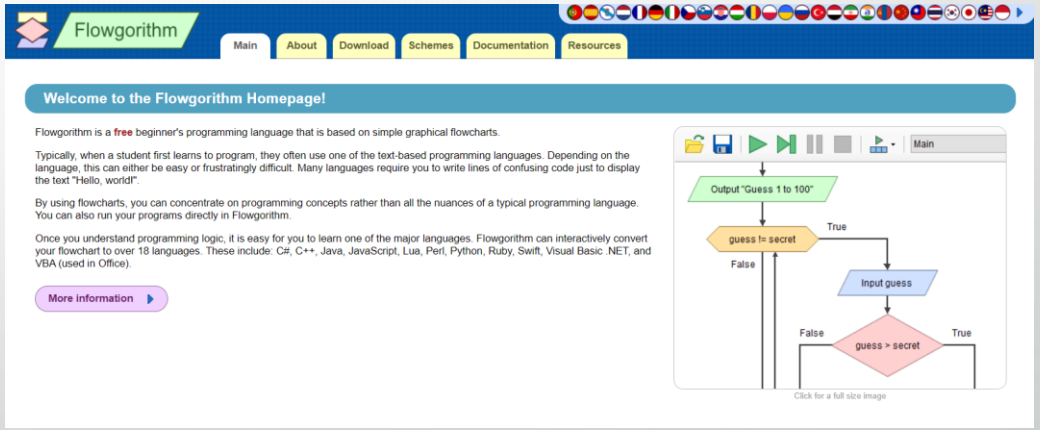
Flow-Chart dell'aliquota progressiva



Nota: Al1, Al2, Al3, R1 e R2 sono COSTANTI (con Nome)

Flowgorithm

- Permette di realizzare flow chart
- Di eseguirli (anche passo passo)
- Di controllare il contenuto delle variabili passo passo (trace table)
- Sviluppato in C#
- Arrivato alla versione 2.21 a settembre 2019
- Gratuito per Win64 e Win32
- http://flowgorithm.org

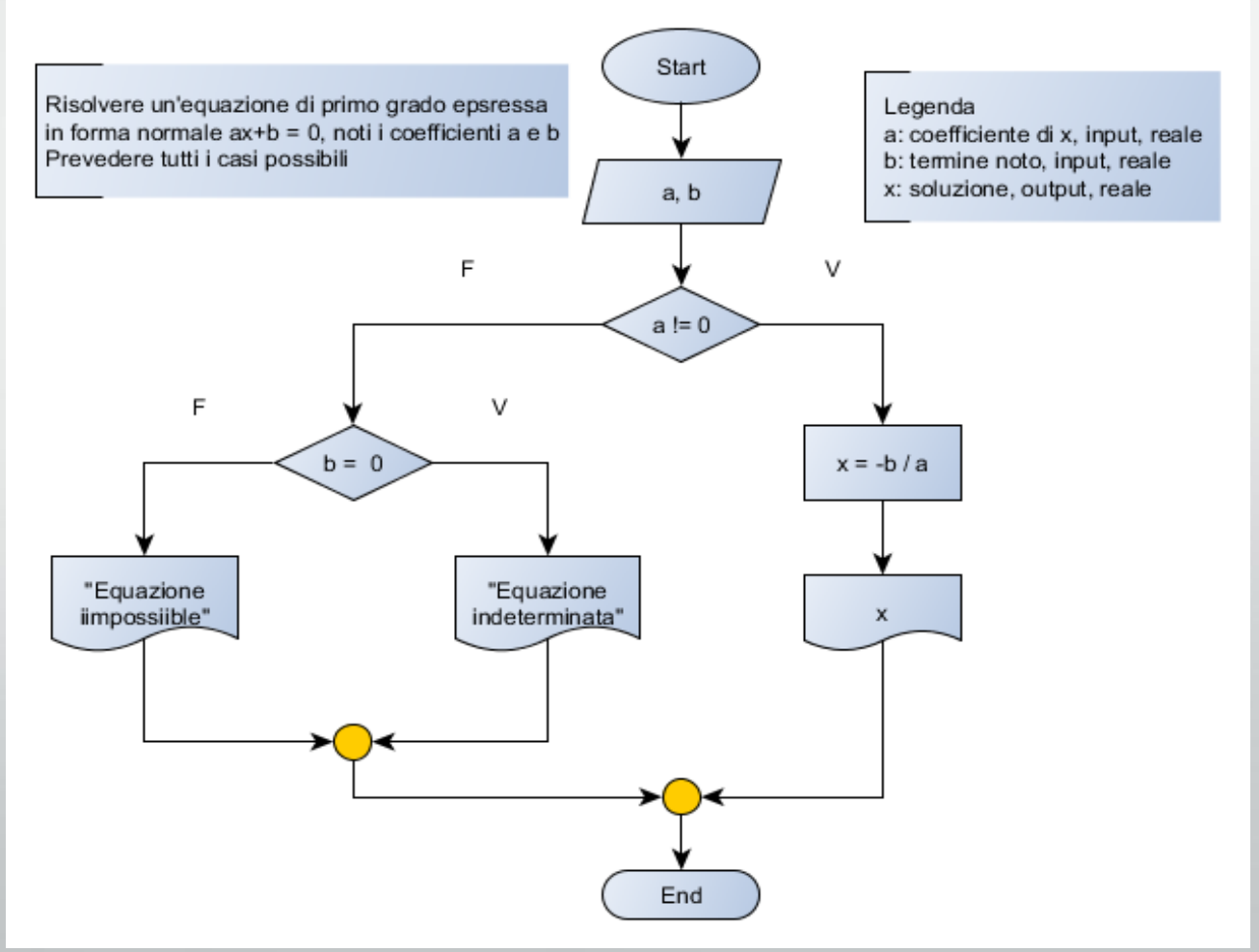


Esr 5: Equazione di 1° grado (1/3)

- Testo: Risolvere un'equazione di primo grado espressa in forma normale
  - `ax + b = 0`,
  - noti i coefficienti `a`, `b`. Prevedere tutti i possibili casi
- Cartella: `05-Eq1Grad`
- https://youtu.be/W8-E8F5pyvQ
- Analisi: I casi possibili sono 3 (determinata, indeterminata, impossibile) quindi occorrono `3-1=2` if.

Esr 5: Equazione di 1° grado (2/3)

Sviluppare il flowchart in Flowgorithm



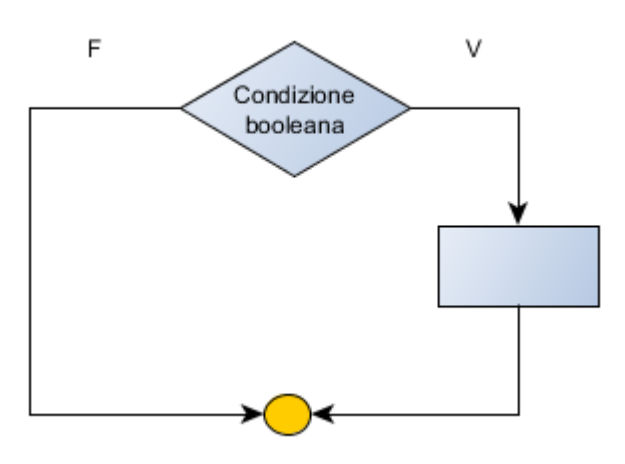
Selezione: teoria

Seleziona

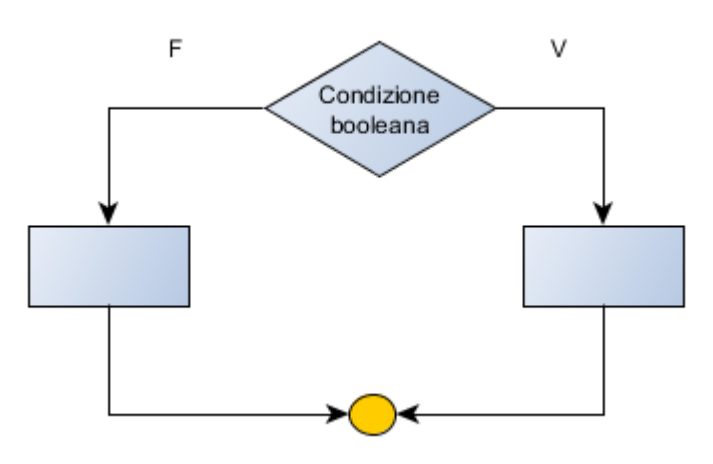
- nomi equivalenti: **Selezione**, **confronto** o **if**
- Ci vuole una **condizione booleana**
  - Un'espressione che dà due possibili risultati: **vero** o **falso**
  - `b + c` NON è un'espressione booleana (è una somma)
  - `a = 3` NON è un'espressione booleana (è un'assegnazione)
  - `a = b * d` NON è un'espressione booleana (è un prodotto seguito da un'assegnazione)

Selezione flowchart

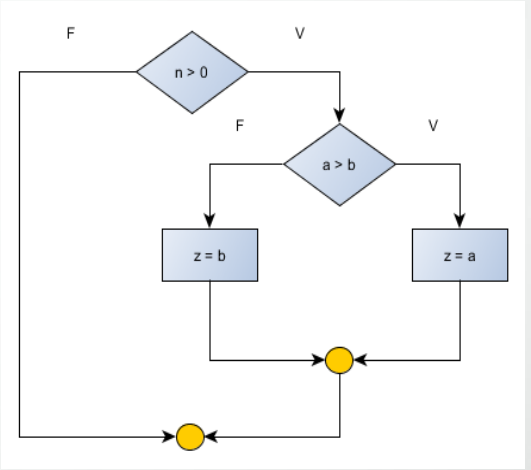
Sia nel flowchart che nella codifica un if deve avere almeno un'istruzione sul ramo vero (mentre sul ramo falso può non averla)



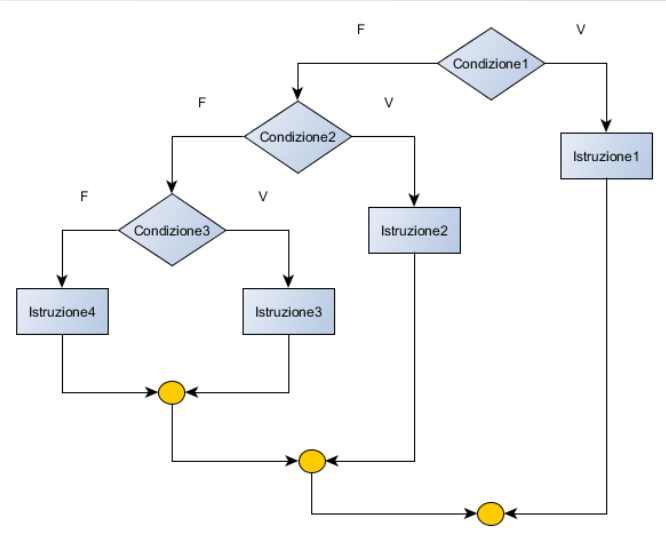
Istruzioni sia sul ramo vero che falso



If innestati



- L'istruzione che possiamo mettere sul ramo vero o falso può essere un qualsiasi gruppo di istruzioni, dunque anche altri if
- Regola fondamentale: prima poi il ramo vero di un if si chiude col suo ramo falso



Operatori relazionali

- **Una condizione booleana è fatta da operatori relazionali**
- **<** minore
- **>** maggiore
- **<=** minore o uguale
- **>=** maggiore o uguale
- **==** uguale
  - Infatti = è l'assegnazione!
  - Nel flowchart possiamo lasciare = tanto è il rombo che ci fa capire che si tratta di una condizione booleana
- **!=** diverso (più raramente in alcuni linguaggi <>)
- Nota sono la stessa cosa
  - `a != b`
  - `! (a == b)` (forma meno adoperata)