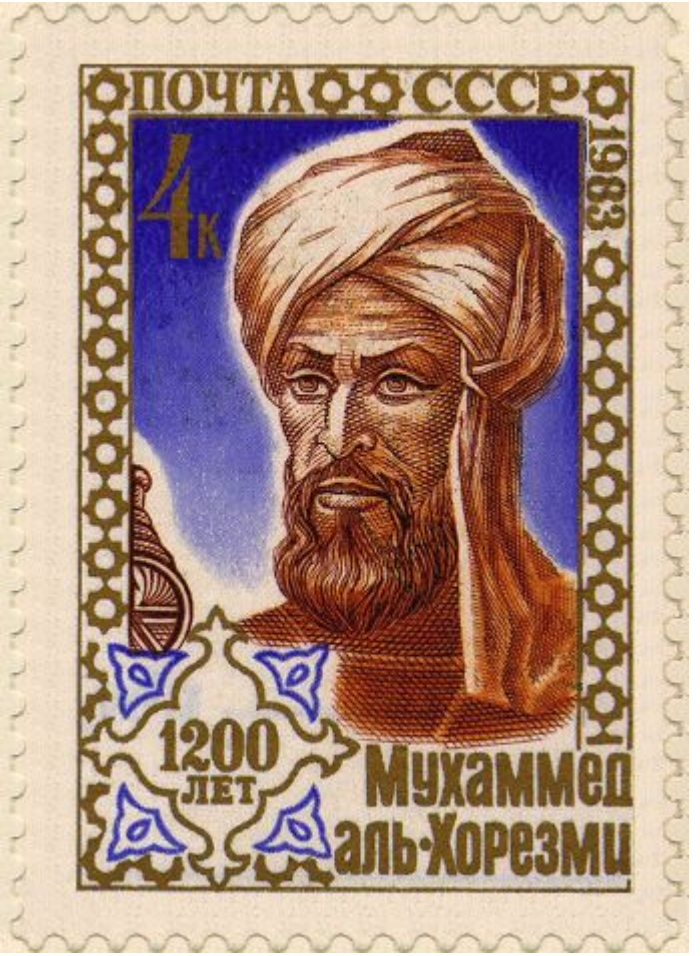
Algoritimi

Moreno Marzolla
Dipartimento di Informatica—Scienza e Ingegneria (DISI)
Università di Bologna
http://www.moreno.marzolla.name/

Algoritmi

- Un algoritmo è un procedimento per risolvere un problema mediante una sequenza finita di passi elementari
- Il termine deriva dal nome del matematico persiano Muḥammad ibn Mūsā al-Khwārizmī (محمد بن موسى الخوارزمي)
  - Autore di un primo fondamentale trattato di algebra (الكتاب المختصر في حساب الجبر والمقابلة al-Kitāb al-mukhtaṣar fī ḥisāb al-jabr wal-muqābala)
  - Un cratere lunare porta il suo nome



Muḥammad ibn Mūsā al-Khwārizmī (c. 780 – c. 850) (francobollo sovietico commemorativo; Fonte: Wikipedia)

Proprietà di un algoritmo

- Atomicità
  - I passi costituenti devono essere elementari, cioè non ulteriormente scomponibili
- Non ambiguità
  - I passi devono essere interpretabili in modo diretto e univoco dall'esecutore, sia esso umano o artificiale
- Finitezza
  - L'algoritmo deve essere composto da un numero finito di passi e richiedere una quantità finita di risorse (es., memoria, ...) per la sua esecuzione
- Terminazione
  - L'esecuzione deve avere termine dopo un tempo finito
- Correttezza
  - L'algoritmo deve produrre il risultato corretto

È un algoritmo?

- Atomicità
- Non ambiguità
- Finitezza
- Terminazione
- Correttezza

"Tutte le qualità di funghi possono fare al caso; ma io ritengo che i porcini sieno da preferirsi, esclusi però i grossissimi. Nettateli bene dalla terra e lavateli, poi tritateli minuti alla grossezza di un cece o anche meno. Metteteli al fuoco con burro, sale e pepe e quando avranno soffritto alquanto, tirateli a cottura con sugo di carne. Ritirati dal fuoco, legateli con balsamella, uova e parmigiano e assodate il composto a bagno-maria. Grammi 600 di funghi in natura con cinque uova faranno uno sformato bastante per dieci persone. Servitelo caldo e per tramesso."

[Pellegrino Artusi (1910), *La scienza in cucina e l'arte di mangiar bene*, 452 Sformato di Funghi]

Cosa si intende per "istruzione elementare"?

- Dipende dall' "esecutore" che consideriamo
- In un esempio di tipo "culinario", le istruzioni devono essere tali da poter essere eseguite in modo non ambiguo da chiunque voglia cimentarsi con la cucina
  - "Separare il tuorlo dell'uovo dall'albume"
  - "Versare due cucchiaini di zucchero nel composto"
  - "Portare l'acqua ad ebollizione"
  - "Frullare il composto per 5 min"
  - ...

Cosa si intende per "istruzione elementare"?

- Noi considereremo come "istruzioni elementari" cose del tipo:
  - Assegnare un valore ad una variabile
    - Es: $a \leftarrow 5$
  - Calcolare il valore di una espressione aritmetica
    - Es: $b \leftarrow (3+a)*5 - 1$
  - Confrontare il valore di due espressioni
    - Es: $a < (b+3)$
  - Chiedere all'utente l'inserimento di un valore, oppure visualizzare un valore
  - ...

Come si descrive algoritmo?

- Linguaggio naturale
- Pseudocodice
- Notazioni grafiche (diagrammi di flusso o *flowchart*)

Esempio: Massimo Comun Divisore
Descrizione in linguaggio naturale

1. Siano *n* e *m* due valori interi positivi
2. Se *n* = *m*, termina indicando *n* come risultato
3. Se *n* > *m*:
   - assegna a *n* il valore (*n* - *m*);
   - torna al punto 2
4. Se *n* < *m*:
   - assegna a *m* il valore (*m* - *n*);
   - torna al punto 2

Vantaggi e svantaggi del linguaggio naturale

- Vantaggi
  - Notazione comprensibile anche ai non esperti
- Svantaggi
  - Il linguaggio naturale può risultare verboso, e quindi produrre una descrizione prolissa
  - Persone diverse hanno stili di scrittura diversi, e questo può generare ambiguità

Esempio: Massimo Comun Divisore

Descrizione mediante *pseudocodice*

Precondizione: quali proprietà devono valere perché l'algoritmo fornisca il risultato desiderato

Postcondizione: quali proprietà valgono al termine dell'esecuzione, nell'ipotesi in cui le precondizioni siano soddisfatte

```
Pre: n > 0, m > 0 entrambi interi
Post: Stampa il MCD(n,m)

    while (n ≠ m) do
        if (n > m) then
            n ← n − m;
        else
            m ← m − n;
        end if
    end while
    Stampa n;
```

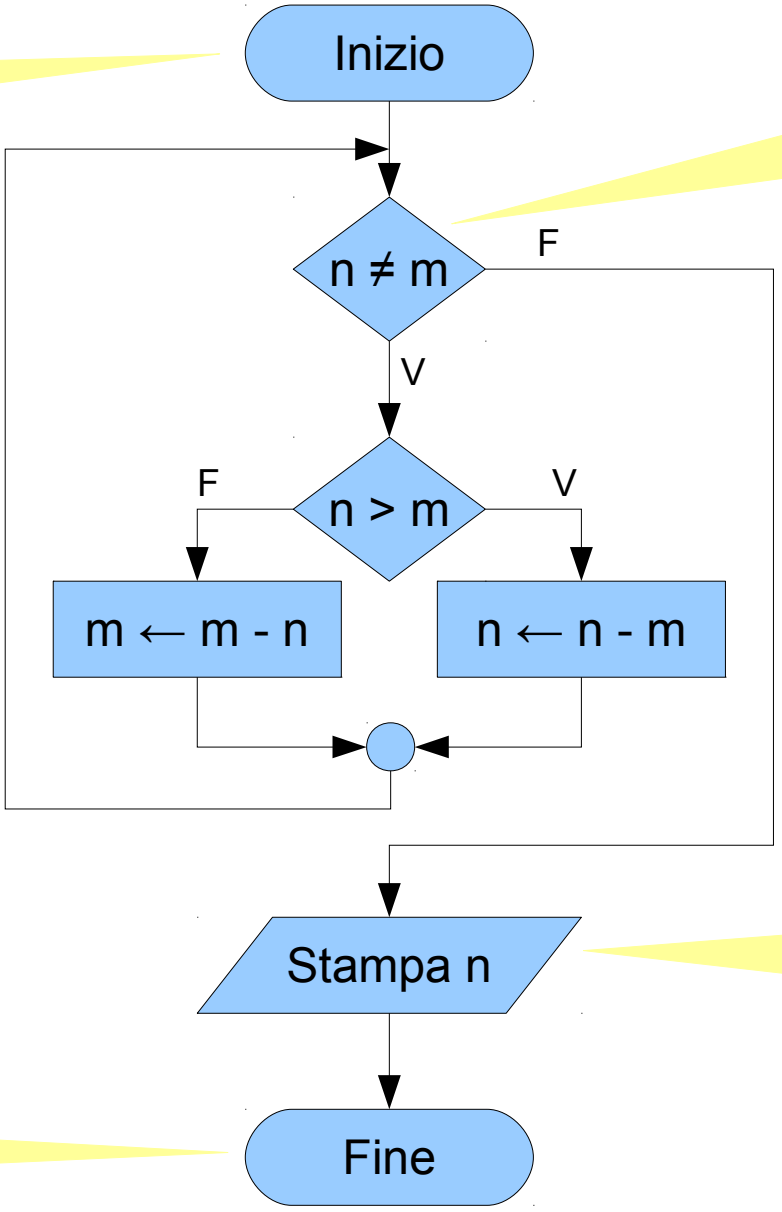
Vantaggi e svantaggi dello pseudocodice

- Vantaggi
  - Più preciso e compatto rispetto al linguaggio naturale
  - Può essere compreso anche da utenti non esperti purché abbiano una minima confidenza con la notazione
  - Facilita la traduzione dell'algoritmo in programma per un "vero" linguaggio di programmazione
- Svantaggi
  - Notazione non univoca: non esiste un unico “tipo” di pseudocodice

Esempio: Massimo Comun Divisore

Descrizione mediante *flowchart*

L'esecuzione inizia da questo nodo



Nodo **condizionale**: deve contenere una espressione che da come risultato vero/falso

Nodo di **input/output**: indica che si chiede l'immissione o la stampa di valori

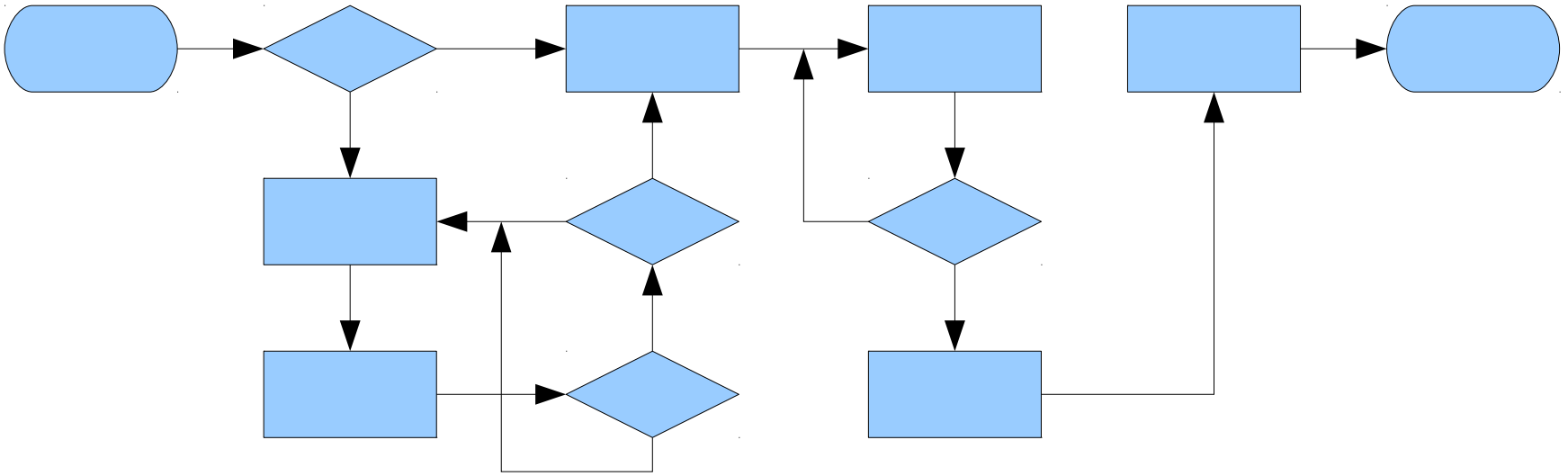
L'esecuzione termina in questo nodo

Vantaggi e svantaggi dei diagrammi di flusso (*flowchart*)

- Vantaggi
  - Notazione di facile comprensione anche per non esperti
  - Consente di individuare “a colpo d'occhio” strutture particolari come cicli e condizioni
  - Esistono software per “disegnare” flowchart e tradurli in programmi
- Svantaggi
  - Possono risultare complessi per algoritmi di dimensioni medio-grandi
    - I flowchart richiedono uno spazio maggiore rispetto alle descrizioni testuali
  - Se abusati, possono produrre descrizioni incomprensibili
  - Non tutti i flowchart possono essere convertiti in programmi strutturati

Flowchart

- In linea di principio, un flowchart può avere una struttura arbitraria...
- ...però questa libertà può portare ad algoritmi difficili (impossibili?) da comprendere



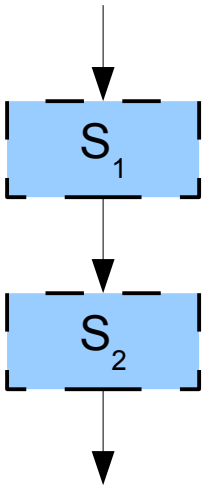
Programmazione strutturata

- Qualsiasi algoritmo può essere espresso combinando tra loro le seguenti strutture di controllo fondamentali
  - Sequenza
  - Condizione
  - Iterazione
- Ognuna di esse ha un singolo punto di ingresso e un singolo punto di uscita
- La programmazione strutturata consente di descrivere algoritmi in modo più comprensibile e meno soggetto a errori
  - Ci sono però casi in cui conviene deviare dalle rigide regole della programmazione strutturata

Costrutti fondamentali della programmazione strutturata

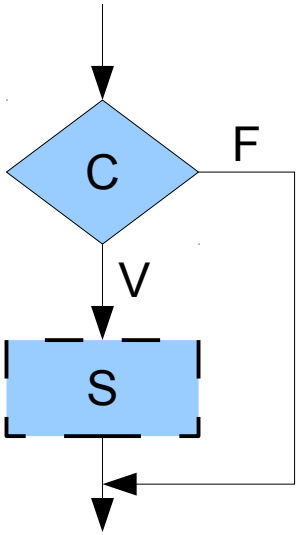
Sequenza

S_1
S_2



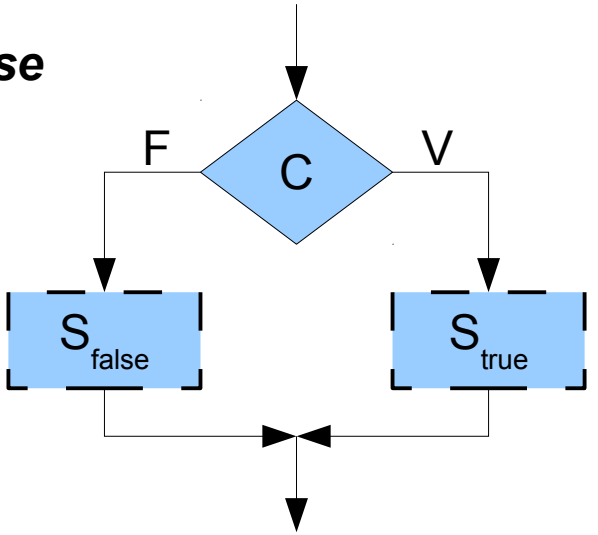
Condizione: *if-then*

```
if C then
    S
end if
```



Condizione: *if-then-else*

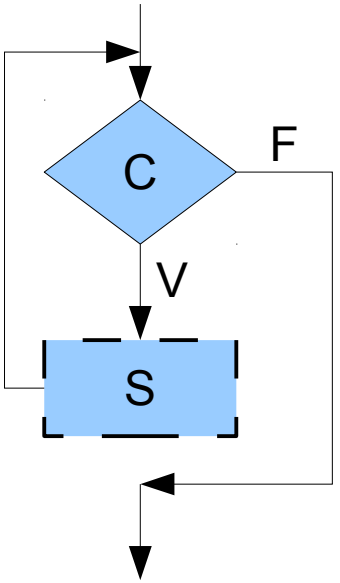
```
if C then
    S_true
else
    S_false
end if
```



Costrutti fondamentali della programmazione strutturata

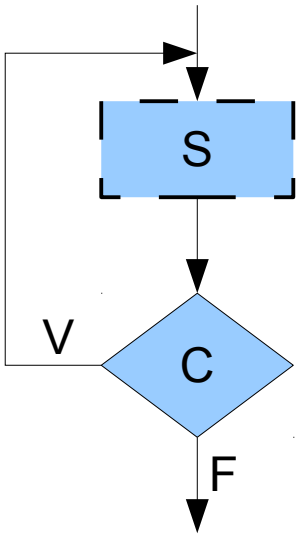
Iterazione: *while-do*

```
while C do
    S
end while
```



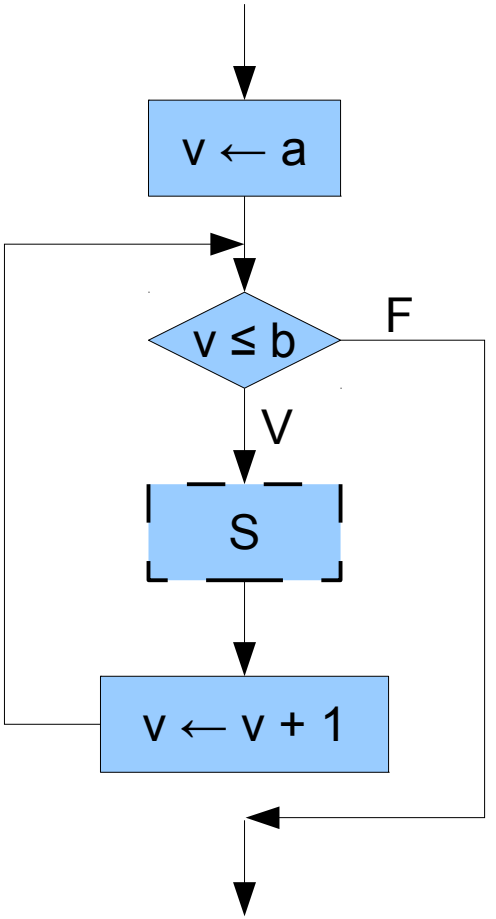
Iterazione: *do-while*

```
do
    S
while C
```



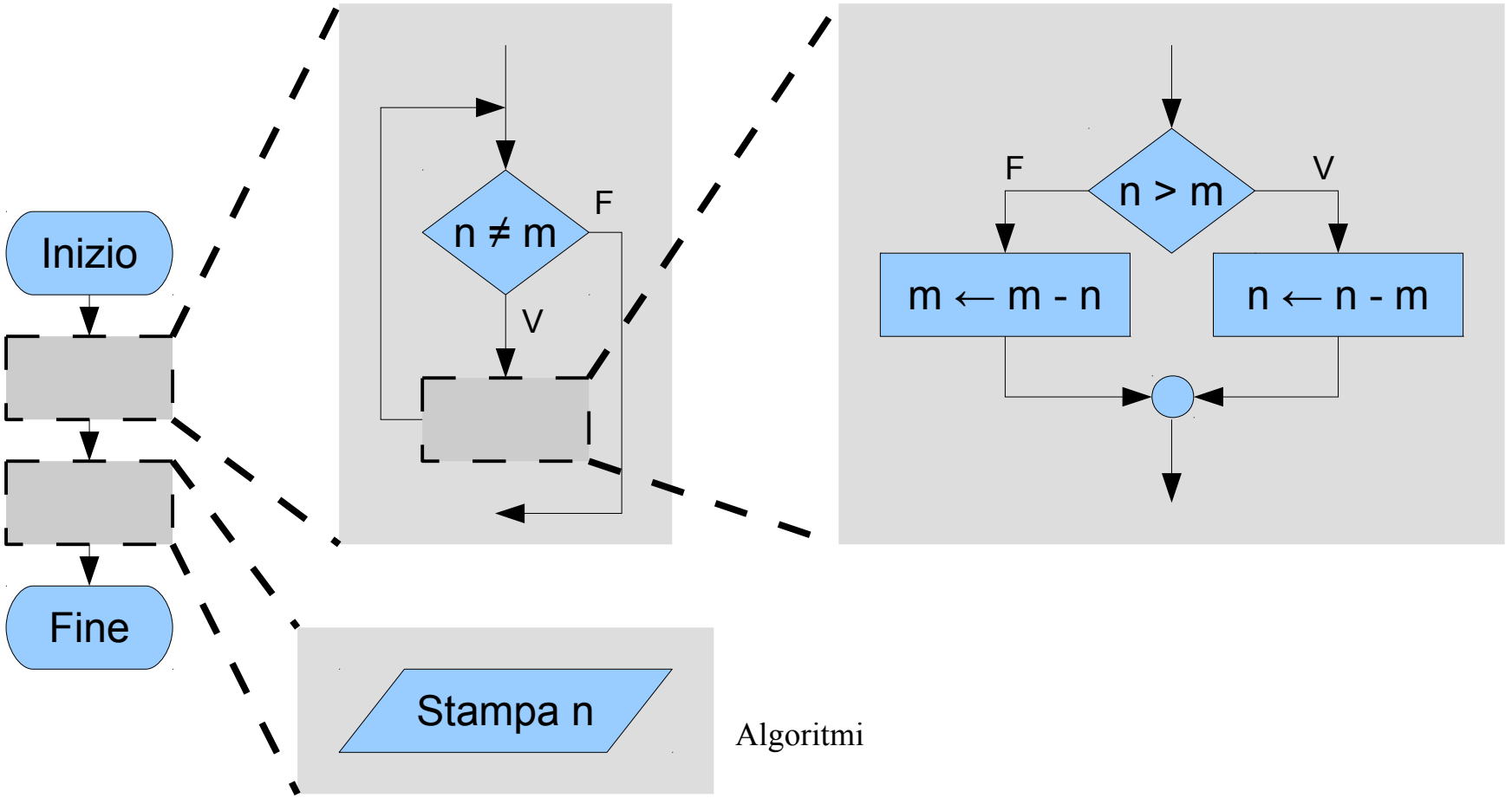
Iterazione: *for*

```
for v ← a to b do
    S
end for
```



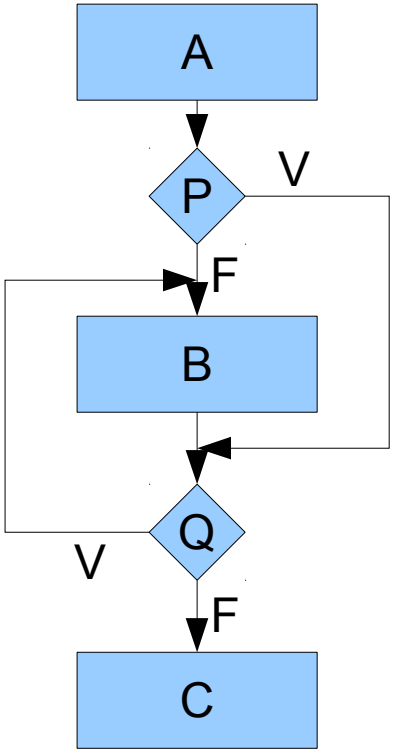
Programmazione strutturata: proprietà di composizione

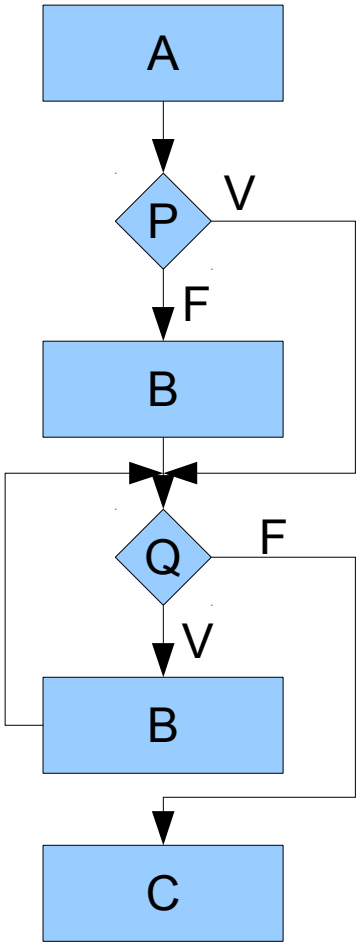
- È possibile rimpiazzare i blocchi tratteggiati con altri costrutti



Attenzione

- Non tutti i diagrammi di flusso possono essere rappresentati con strutture sequenziali, if-then. if-then-else, do-while, while-do e for
- Esempio:






```
A
if (not P) then
    B
end if
while (Q) do
    B
end while
C
```

Esempio: Elevamento a potenza

- Scrivere un algoritmo che, dati in input un numero reale $x \neq 0$ e un intero $n \geq 0$, calcola il valore x^n

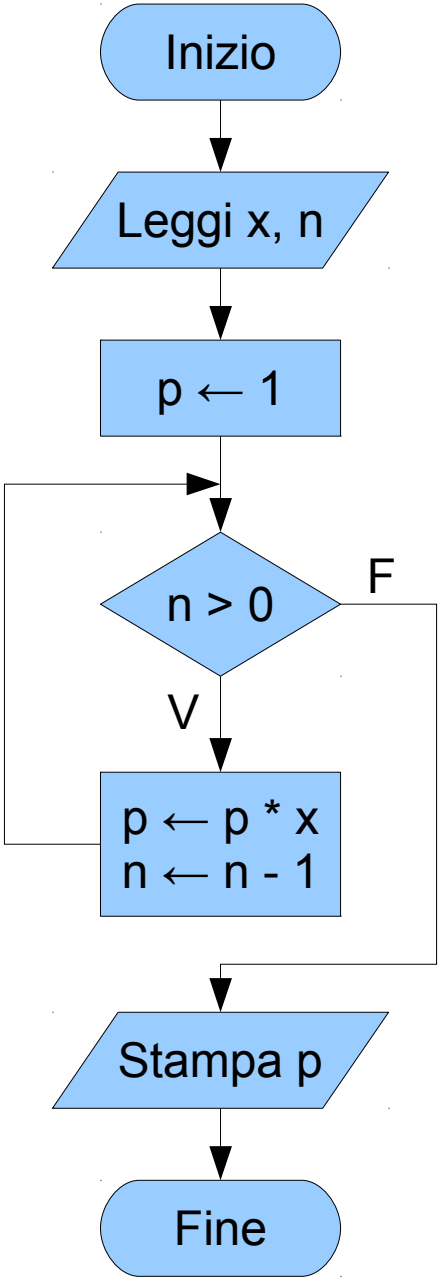
$$x^n = \underbrace{x \times x \ldots \times x}_{n \text{ volte}}$$

  - Per definizione si ha che $x^0 = 1$

Elevamento a potenza

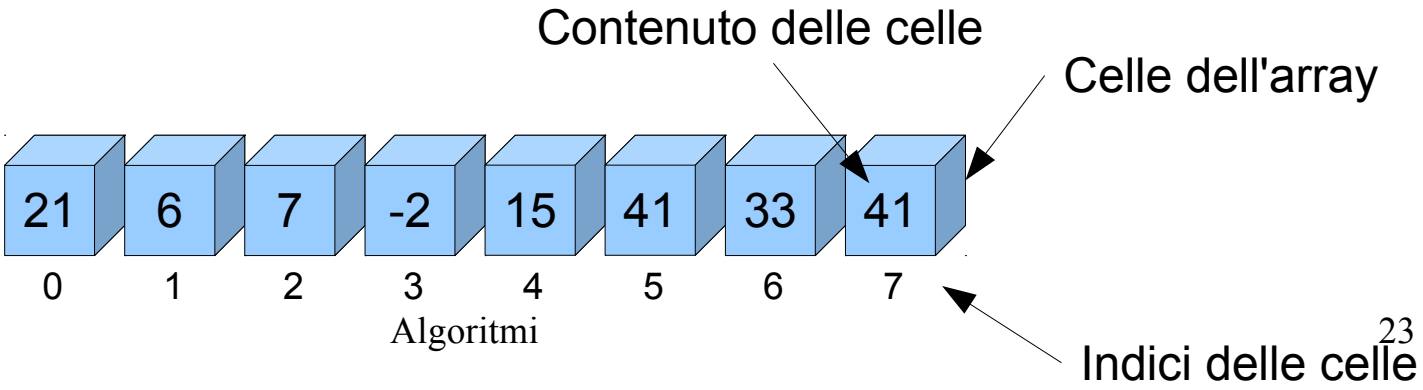
Pre: x ≠ 0, n intero ≥ 0
Post: Stampa p = x^n

```
p ← 1
while (n > 0) do
    p ← p * x
    n ← n – 1
end while
Stampa p
```



Esempio: Massimo e Minimo

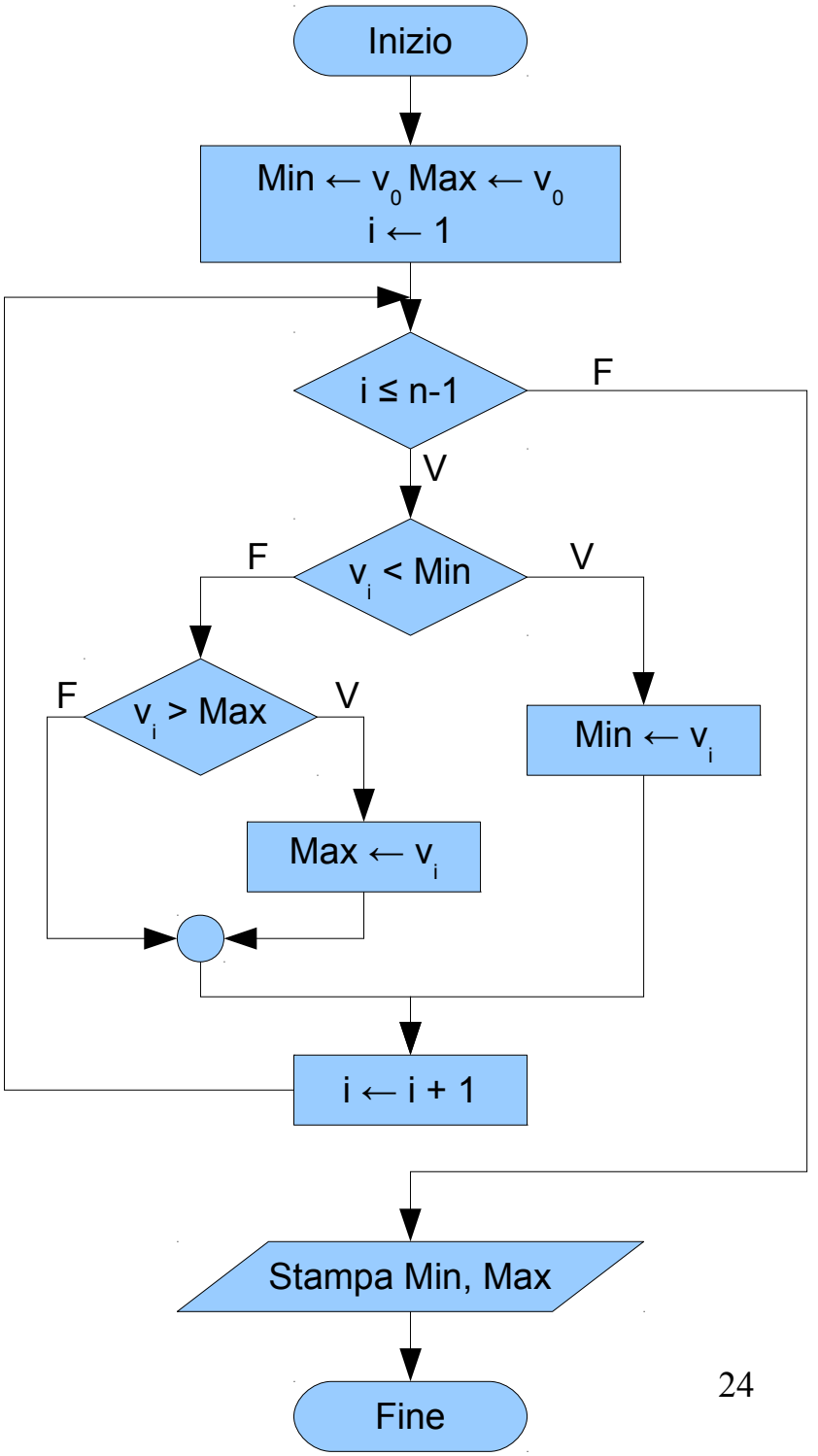
- Dato un *array* (vettore) non vuoto di valori reali v_0, v_1, … v_{n-1}, scrivere un algoritmo per determinare il valore massimo e minimo
  - Se tutti i valori sono uguali, massimo e minimo coincidono
  - Se l'array ha un solo elemento (*n* = 1), massimo e minimo coincidono con l'unico valore presente
- Un array di *n* elementi può essere intuitivamente immaginato come un insieme di *n* "scatole" (celle), ciascuna etichettata con un intero da 0 a *n* – 1 e contenente un valore



Possibile soluzione

Pre: $v_0, v_1, \ldots v_{n-1}$ array non vuoto di valori reali ($n \geq 1$)
Post: Min, Max sono i valori minimo e massimo presenti nell'array

```
Min ← v0
Max ← v0
for i ← 1 to n-1 do
    if (vi < Min) then
        Min ← vi
    else if (vi > Max) then
        Max ← vi
    end if
end for
Stampa Min, Max;
```



Esercizio per casa
Ricerca sequenziale

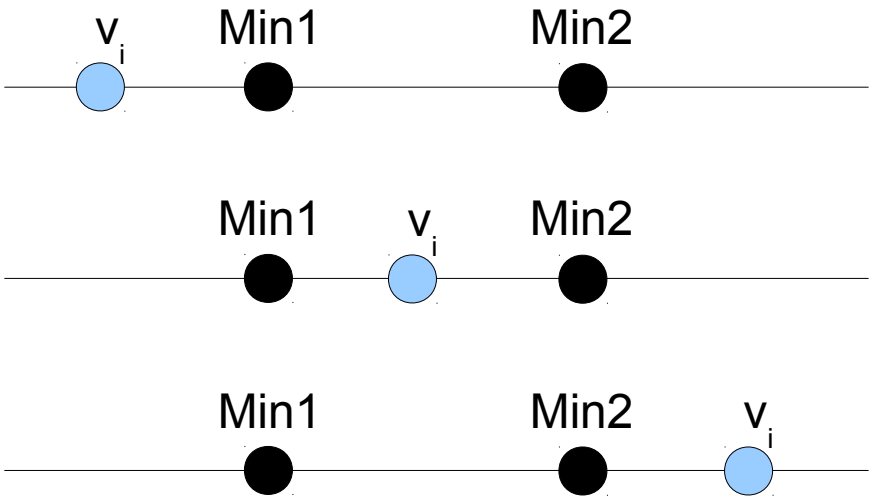
- Dati:
  - Un array di valori reali v_0, v_1, … v_{n-1} non necessariamente distinti e non ordinati
  - Un valore reale *x* (arbitrario)
- Determinare la posizione di una occorrenza di *x* nell'array (se presente)
  - Cioè determinare un indice *i*, se esiste, tale che $v_i = x$
  - Se esistono più elementi dell'array con valore *x* è possibile restituire l'indice di una qualunque occorrenza
  - Se *x* non compare nell'array, restituire -1

Esercizio per casa
Primo e secondo minimo

- Dato un array di valori reali distinti $v_0, v_1, \ldots v_{n-1}$ con almeno due elementi ($n \geq 2$), scrivere un algoritmo per determinare i due valori minimi
- Es:
  - 1, 9, -3, 7, 5, 12, 4 → ritorna -3, 1
  - 1, 2, 3, 4, 5 → ritorna 1, 2

Primo e secondo minimo

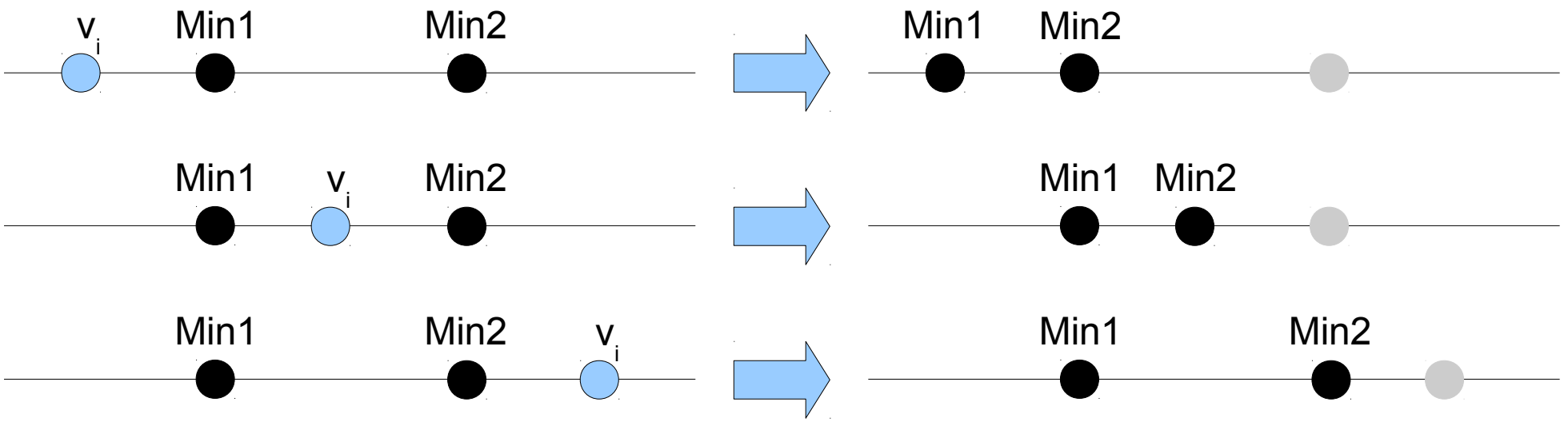
- Idea: esaminiamo uno per uno gli elementi dell'array
- Supponiamo di aver determinato i valori Min1, Min2 del primo e secondo minimo del sottovettore v_1, v_2, … v_{i-1}
- Esaminiamo il valore v_i
- Tre casi possibili
  - v_i < Min1
  - Min1 < v_i < Min2
  - v_i > Min2



(Per ipotesi tutti i valori sono distinti, quindi non si potrà mai verificare che v_i = Min1 oppure v_i = Min2)

Primo e secondo minimo

- In ciascuno dei tre casi, come dobbiamo modificare i valori Min1 e Min2 affinché essi siano ancora il primo e il secondo minimo del sottovettore $v_0, v_1, \ldots v_i$?



- Come definiamo Min1 e Min2 all'inizio?

Esercizio per casa
Deposito

- Il primo gennaio dell'anno zero, Augusto deposita l'equivalente di 1 Euro in banca, negoziando un tasso composito del 5% annuo
  - Il primo gennaio dell'anno 1 si trova nel conto $1\times(1+0.05) = 1.05$ Euro
  - Il primo gennaio dell'anno 2 si trova nel conto $1.05\times(1+0.05) = 1.1025$ Euro
  - Il primo gennaio dell'anno 3 si trova nel conto $1.1025\times(1+0.05) = 1{,}157625$ Euro
  - … e così via
- Il primo gennaio dell'anno zero, Claudia deposita l'equivalente di 100 Euro in banca, negoziando un tasso composito del 4% annuo
- Entrambi i depositi vengono tramandati di generazione in generazione, alle stesse condizioni
- In quale anno l'importo presente al primo gennaio nel conto (degli eredi) di Augusto diventa strettamente maggiore di quello presente nel conto (degli eredi) di Claudia?