{ELEMENTI DI INFORMATICA}

APPUNTI PER IL CORSO DI INFORMATICA MUSICALE

INDICE DEI CONTENUTI:

1. PARTE PRIMA

Digitale: tutto ciò che viene rappresentato con numeri. In forma digitale, un insieme di informazioni viene rappresentato come una sequenza di numeri presi da un insieme di valori discreti… digitale come espresso riferimento alla matematica del discreto.
Analogico: tutto ciò che non è numerabile, non rappresentabile entro un insieme discreto di valori.

Un po' di storia

Non è affatto recente la necessità di poter contare su strumenti in grado di calcolare ed elaborare dati velocemente e senza errori. E, anche se le prime testimonianze di calcolo automatico fanno riferimento a ritrovamenti (in Mesopotamia) di tavole babilonesi risalenti al 1800 – 1600 a.C., si può effettivamente parlare di macchine calcolatrici solo nel XVII sec. Scienziati del calibro di Pascal e Leibniz furono i primi ad affrontare il problema dell'automatizzazione del ragionamento logico-matematico sviluppando macchine, fatte di sole parti meccaniche, in grado di eseguire operazioni come la somma e la sottrazione.
Nel 1833, Babbage concepisce la prima macchina meccanica programmabile per tabulare funzioni polinomiali, la *macchina alle differenze* (differenze engine).
Nel 1890 H. Hollerith sviluppa una *macchina a schede perforate* per eseguire statistiche del censimento degli Stati Uniti. Funzionava in questo modo: i dati venivano immessi su schede di cartone perforate in un certo modo; successivamente, venivano contate da una specie di pantografo che permetteva elaborazioni del tipo somma, media, etc…
Nel 1932, K. Zuse realizza la prima macchina elettromeccanica in grado di eseguire calcoli programmati ed introduce il sistema di numerazione binario.
Durante la seconda guerra mondiale, il matematico Turing costruisce il primo elaboratore elettronico della storia per decodificare i messaggi che i tedeschi criptavano con la macchina Enigma.
Nel 1950, il matematico J. Van Neumann realizzò il primo calcolatore a programma memorizzato, l'EDVAC, che pesava otto tonnellate e aveva una memoria di 1024 parole.
Lo schema della sua macchina è praticamente il modello su cui si basano tutti i moderni computer. È suddiviso in quattro elementi fondamentali:

1. la memoria;
2. l'unità aritmetico logica (ALU);
3. l'unità di controllo;
4. le unità di ingresso/uscita (I/O)

Il punto di svolta nell'evoluzione? L'invenzione del transistor, un dispositivo che opera assumendo solo due stati (binario): 1-acceso, 0-spento. È, la miniaturizzazione delle valvole termoioniche (per intenderci, le componenti della macchina di Touring).

Informatica e calcolatore

L'informatica è lo studio sistematico degli algoritmi che descrivono e trasformano l'informazione: la loro teoria, analisi, progetto, efficienza, realizzazione (ACM – Association for Computer Machinery).
È, dunque, la scienza della rappresentazione e dell'elaborazione automatica dell'informazione.

Un calcolatore è, invece, un'apparecchiatura digitale, elettronica ed automatica capace di effettuare trasformazioni sui dati. *Digitale*, perché i dati sono rappresentati mediante un alfabeto finito fatto di sole cifre (*digit*); *elettronica*, perché realizzata l'uso di tecnologie elettroniche e, infine, *automatica* perché in grado di eseguire una successione di operazioni senza alcun intervento esterno.
È fondamentale sottolineare che, la capacità di eseguire una serie di operazioni in modo del tutto automatico è possibile solo grazie ad un dispositivo di *memoria*, in cui vengono registrati i dati e la descrizione delle operazioni da eseguire nell'ordine in cui devono essere effettivamente avvenire (*programma*).

- PROGRAMMA: letteralmente, una sequenza di operazioni che preparano l'elaboratore alla soluzione di un determinato problema, descrive un algoritmo in modo che l'elaboratore possa comprenderlo;
- ALGORITMO: una sequenza finita di istruzioni per la risoluzione di un certo problema.

Il calcolatore è, una macchina cosiddetta *universale*: basta cambiare il programma in memoria per risolvere problemi di natura differenti.

L'algoritmo

…una sequenza finita step che portano alla soluzione del problema. Nello specifico: *un insieme ben ordinato di operazioni assolutamente non ambigue ed effettivamente calcolabili che, applicate ad un insieme di condizioni iniziali, produce un risultato e termina in una quantità di tempo finita (nel minor tempo possibile ed occupando poca memoria)*[1].

Le proprietà necessarie di un algoritmo sono:

- ***atomicità***: gli step devono essere elementari, non ulteriormente scomponibili;
- ***non ambiguità***: gli step devono essere interpretabili in modo univoco;
- ***finitezza***: l'algoritmo deve essere composto da un numero finito di step e richiedere un numero finito di dati in ingresso;
- ***terminazione***: l'esecuzione deve terminare dopo un tempo finito;
- ***effettività***: deve portare ad un risultato univoco;

[1] Cfr. F. Cattadori, *Appunti di Informatica*

- ***determinismo***: ogni step deve essere chiaramente stabilito.

Facciamo un esempio: supponiamo di voler determinare il maggiore tra due numeri x e y.

step 1: esegui la differenza tra x e y $[d \longleftarrow (x - y)]$
step 2: valuta se d è maggiore di zero:

- se tale condizione è vera, allora la soluzione è x;
- se tale condizione è falsa, allora la soluzione è y.

Hardware e software

Con *hardware* si intende, letteralmente, la parte fisica del PC; con *software*, la parte logica, i programmi che fanno girare le singole componenti hardware.

La componente hardware principale è la *scheda madre*, che a sua volta si compone di:

- processore CPU: ha il compito di eseguire le istruzioni di un programma e sovrintende al funzionamento dell'intera macchina. Esegue tutti i calcoli, gestisce il trasferimento dei dati attraverso le memorie e attiva o disattiva i componenti della macchina. Per poter fare quanto detto, una cpu esegue quattro fasi:

    - **fetching:** nella prima fase recupera dalla memoria tutti i dati necessari per eseguire l'operazione. Detti dati vengono copiati nella memoria interna al processore, la *cache*;
    - **decoding:** in questa seconda fase, decodifica tutti i dati in linguaggio binario, in modo da poter essere compresi dalla cpu e, ne valuta l'ordine di esecuzione;
    - **executing:** raccoglie tutti i dati e li elabora con complesse operazioni matematiche in modo da ritornare ciò che gli è stato richiesto;
    - **write back:** terminata la fase di esecuzione, il risultato viene copiato nella memoria per passare all'istruzione successiva.

  La velocità del processore dipende dal numero di cicli di istruzioni al secondo che può eseguire (un intero ciclo si compone delle quattro fasi appena esposte).
  Il ritmo di lavoro di un processore è scandito da un segnale elettrico detto *clock* (rapidissimi impulsi che si ripetono un enorme numero di volte al secondo). La velocità di clock si misura in GHz (giga hertz).

- memoria RAM: memoria ad accesso casuale, un tipo di memoria volatile. È qui che vengono caricati i programmi che la CPU deve eseguire. Si dice volatile, perché una volta chiuso il programma, le informazioni immagazzinate vengono definitivamente

cancellate. Si potrebbe immaginare la memoria RAM come una sorta di recipiente provvisorio per tutti quei dati di cui la macchina necessita in quel preciso istante. Fondamentale è la *frequenza*, espressa in Mhz (mega hertz), che misura proprio la frequenza con cui queste memorie aggiornano i dati.
Altro elemento che gioca un ruolo importantissimo nelle prestazioni di una memoria RAM è la *latenza*, ovvero, il ritardo in cui è inoltrato un segnale di richiesta in lettura e il momento in cui il dato è effettivamente pronto per essere restituito: minore latenza, maggiori prestazioni! Si misura in CL.

- disco rigido HD ed SSD: è un dispositivo di archiviazione, un tipo di memoria di massa. Solitamente, per la memorizzazione dei dati, un disco rigido necessita di una operazione preliminare di formattazione logica e scelta del sistema di archiviazione o *file system*. Fisicamente, i dati sono scritti su disco secondo un preciso modello di allocazione. Il più diffuso è sicuramente il CHS o *cilindro-testina-settore*: i dati hanno un preciso indice di cilindro, di testina e di settore.

  Senza troppo dilungarci, un'unità di memoria allo stato solido o SSD, a differenza di un HD, ha la possibilità di memorizzare in modo permanente grandissime quantità di dati senza dover ricorrere all'utilizzo di organi meccanici (testine, piatti, etc…).

2. PARTE SECONDA

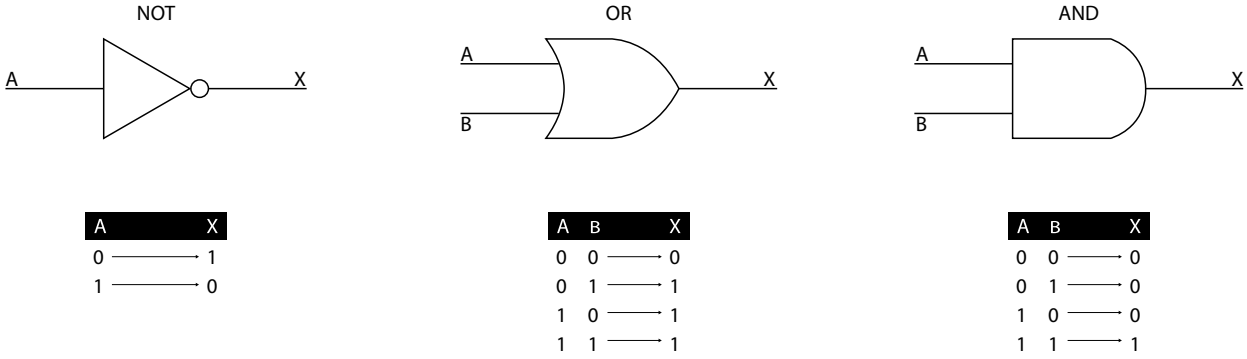
Algebra di Bool

I circuiti logici sono componenti hardware che gestiscono informazioni binarie, quelli di base sono detti *porte logiche*. Potremmo immaginare il calcolatore, allora, come una complessa rete logica, un insieme di porte logiche opportunamente connesse, che può essere descritto per mezzo di una particolare notazione matematica che specifica lo stato di ogni porta, parliamo dell'algebra di Bool o booleana.

Le funzioni dell'algebra booleana sono isomorfe ai circuiti digitali: un circuito può essere espresso tramite un'espressione booleana e viceversa.

Contempla dolo due costanti: 0 e 1, falso e vero ed è basata su tre operatori: *AND, OR e NOT*:

- *l'operatore AND*: si definisce operatore di prodotto logico. Il valore del prodotto logico è 1 se il valore di tutti gli operandi è il simbolo 1;
- *l'operatore OR*: si definisce operatore di somma logica. Il valore della somma logica è 1 se almeno uno degli operandi è il simbolo 1;
- *l'operatore NOT*: si definisce operatore di negazione. Inverte il valore della costante su cui si opera.



Tutte le funzioni booleane possono essere espresse come combinazione di questi tre operatori!

Sistemi di numerazione in base B

La base di un sistema di numerazione posizionale è il numero di cifre, inclusa quella per lo zero, che lo stesso sistema usa per rappresentare i numeri.
La cifra di minor valore è sempre lo zero, mentre le altre sono nell'ordine 1, 2, 3, …B-1.
Un numero intero x si rappresenta come $(c_n c_{n-1} \dots c_2 c_1 c_0)_B$

$$x = c_n B^n + c_{n-1} B^{n-1} \dots c_2 B^2 + c_1 B^1 + c_0 B^0$$

…da destra verso sinistra, la cifra meno significativa è c_0, quella più significativa c_n. Allo stesso modo, un numero frazionario y $(0, c_1 c_2 \dots c_n)_B$

$$x = c_1 B^{-1} + c_2 B^{-2} \dots c_n B^{-n}$$

con n cifre in base B si rappresentano tutti numeri interi positivi da 0 a $B^n - 1$.
Ad esempio, in base 10:

$$da\ 0\ a\ 10^2 - 1 = 99$$

numeri a due cifre da zero a novantanove.

Sistema binario, bit e byte

La base 2 è la più piccola base per un sistema di numerazione, con sole due cifre: 0 e 1 (bit-**bi**nary digi**t**).

Consideriamo la sequenza binaria $(010111)_2$ e calcoliamo il numero espresso.

$$(01011)_2 = 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

$$= 0 + 16 + 0 + 4 + 2 + 1$$

$$= (23)_{10}$$

L'elemento alla base della rappresentazione delle informazioni nei calcolatori è il byte… nient'altro che un numero binario ad otto cifre (8 bit). I numeri interi rappresentabili con 1 byte vanno da 0 a $2^8 - 1 = 255$.

Sistema esadecimale

Altra base largamente usata in campo informatico è la base 16.

$$0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ A\ B\ C\ D\ E\ F$$

…oltre il 9, la corrispondenza in decimale è:

$$A = (10)_{10} \quad B = (11)_{10} \quad C = (12)_{10}$$
$$D = (13)_{10} \quad E = (14)_{10} \quad F = (15)_{10}$$

Esempio: il valore $(9F1C)_{16}$ corrisponde a

$$9 \cdot 16^3 + 10 \cdot 15^2 + 1 \cdot 16^1 + 10 \cdot 12^0 =$$

$$= 36864 + 2250 + 16 + 1$$

$$= (39131)_{10}$$

Una cifra esadecimale corrisponde a 4 bit…

0	0000	|	1000	8
1	0001	|	1001	9
2	0010	|	1010	A
3	0011	|	1011	B
4	0100	|	1100	C
5	0101	|	1101	D
6	0110	|	1110	E
7	0111	|	1111	F

In questo modo è possibile rappresentare numeri binari molto lunghi con poche cifre!
32 bit, ad esempio corrispondono ad 8 cifre esadecimali.
Il procedimento per convertire dal sistema esadecimale a quello binario è estremamente semplice, basta espandere ciascuna cifra con i quattro bit corrispondenti. Proviamo con il numero 0x0f7a (0x… si usa per rappresentare numeri esadecimali):

0	F	7	A
0000	1111	0111	1010

Il valore corrispondente? Un numero binario di 16 bit ($4 \cdot 4$):

$$(0000111101111010)_2$$

3. PARTE TERZA

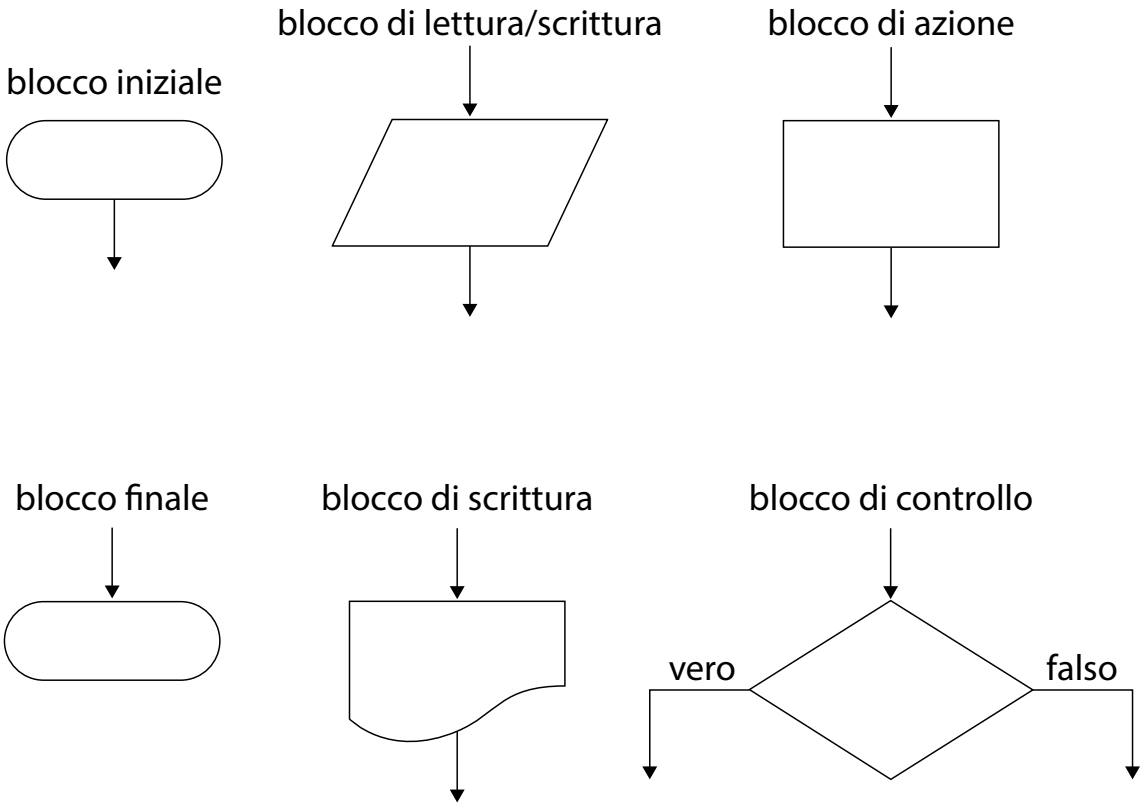
Rappresentare gli algoritmi: i diagrammi di flusso

Un algoritmo è una sequenza finita ed ordinata di operazioni; un *diagramma di flusso* (in informatica, sinonimo di *diagramma a blocchi*) è la rappresentazione grafica di un algoritmo.

…un diagramma a blocchi descrive il flusso delle operazioni da eseguire per realizzare la trasformazione, definita nell'algoritmo, dai dati iniziali ai risultati.

Un algoritmo, prima di essere passato al calcolatore, può essere rappresentato graficamente con un diagramma a blocchi, in modo da avere una visione chiara di tutte le istruzioni necessarie per l'implementazione.

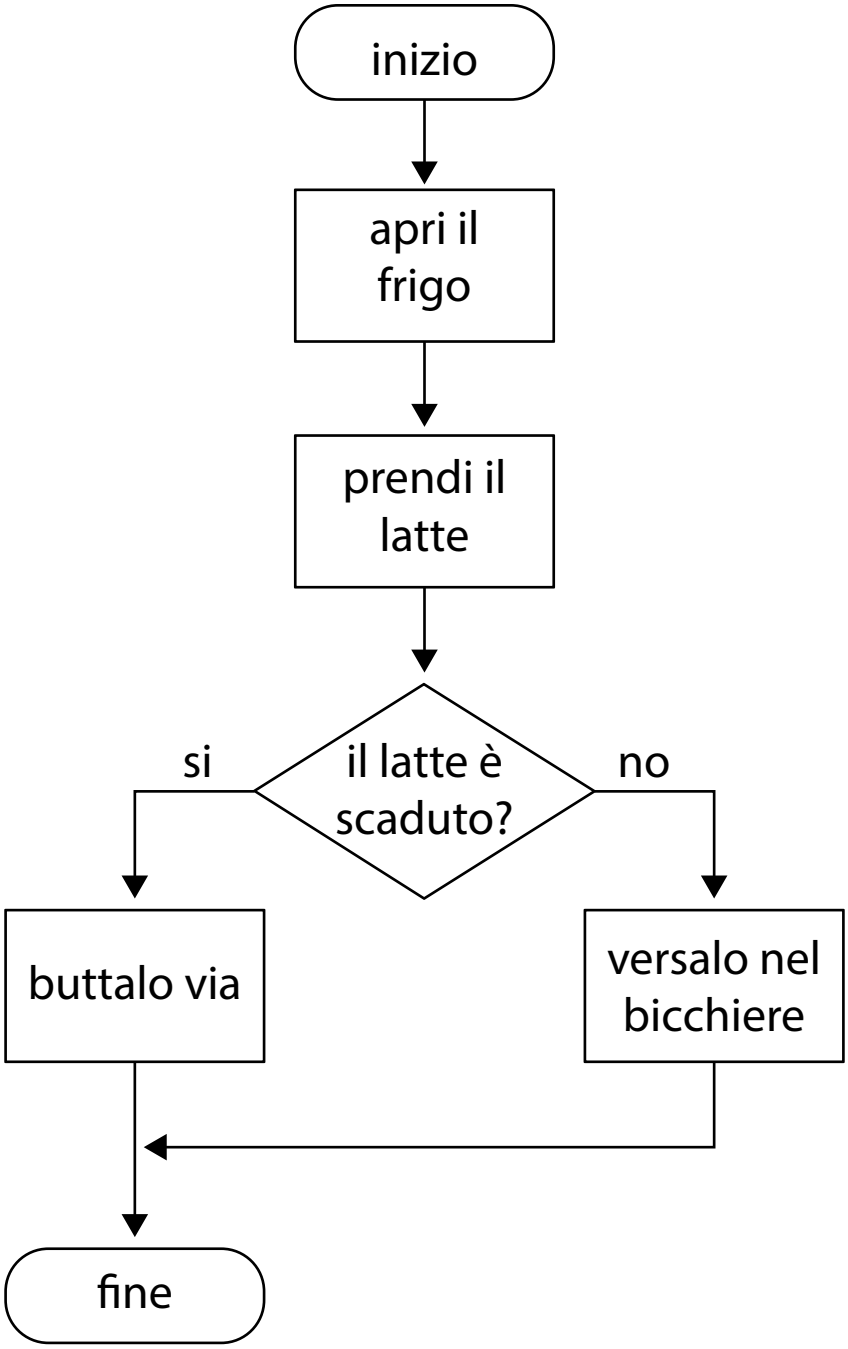
Un diagramma di flusso (*flow chart*) utilizza forme geometriche semplici o *blocchi elementari*, la cui forma è determinata dal tipo di istruzione, collegate tra loro da *linee di flusso frecciate* che rappresentano il susseguirsi di azioni elementari.



blocco iniziale: rappresenta l'inizio dell'algoritmo;
blocco finale: la fine dell'algoritmo;

blocco di lettura/scrittura: rappresenta un'istruzione di comunicazione, lettura o scrittura di un dato;
blocco di scrittura: un'istruzione di comunicazione scrittura di un dato;
blocco di azione: un'azione, ad esempio il calcolo di una determinata funzione;
blocco di controllo: rappresenta un'istruzione di controllo. Esempio, controllare se un valore è pari o dispari.

Facciamo un semplice esempio: *prendi il latte dal frigo, controlla la scadenza e se il latte è ancora buono versalo nel bicchiere, altrimenti buttalo via*.



È importante sottolineare che:

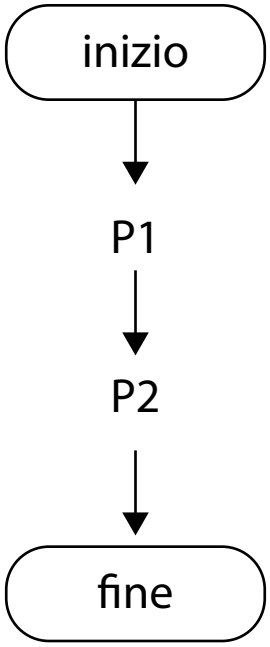
- ciascun blocco di azione o di lettura/scrittura ha una sola freccia entrante ed una sola uscente;
- ciascun blocco di controllo ha una sola freccia entrante e due uscenti;
- ciascuna freccia entra in un blocco o s'innesta in un'altra freccia;
- ciascun blocco è raggiungibile dal blocco iniziale;
- il blocco finale è raggiungibile da qualsiasi altro blocco.

Analisi strutturata: schema sequenziale, di selezione e di iterazione

L'analisi strutturata consente la descrizione di algoritmi facilmente documentabili e comprensibili.

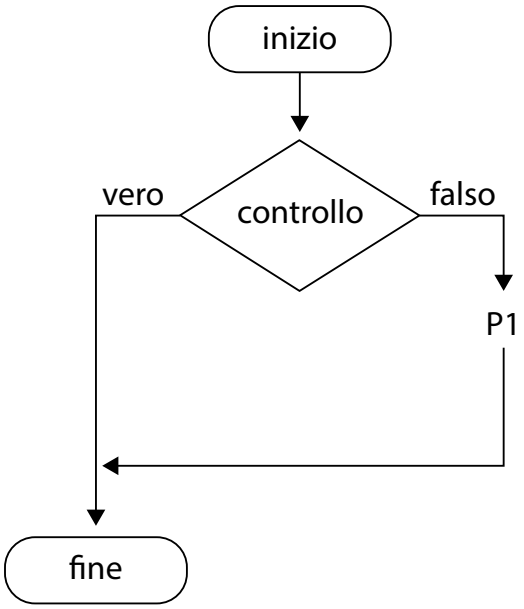
Una descrizione è di tipo strutturato se i blocchi sono collegati secondo i seguenti schemi di flusso strutturati:

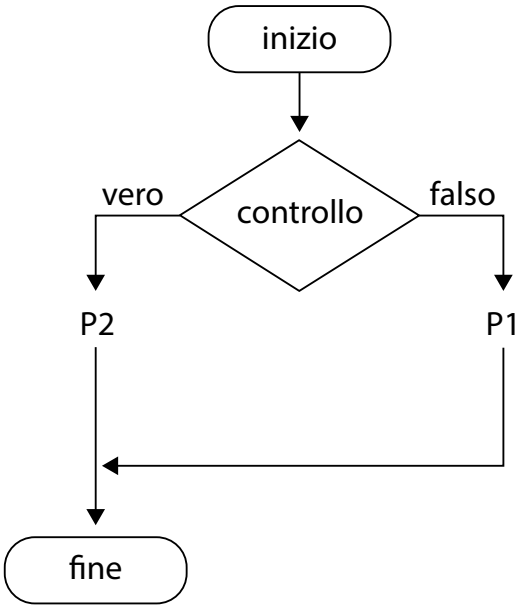
1. SCHEMA DI SEQUENZA: collegati in sequenza. Due o più schemi sono eseguiti in sequenza.



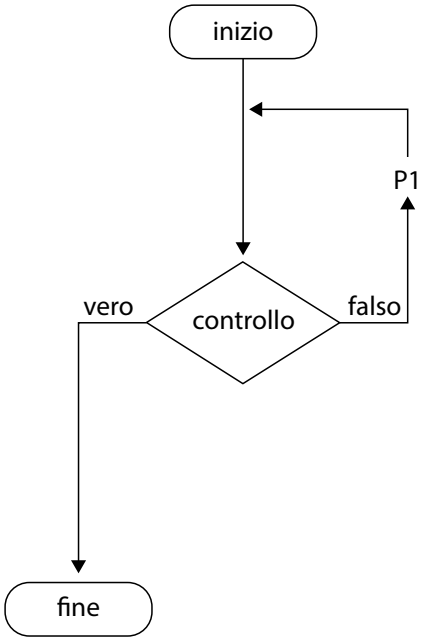
   Lo schema di sequenza è strutturato solo se lo sono i blocchi P1 e P2!

2. SCHEMA DI SELEZIONE: un blocco di controllo subordina l'esecuzione di due possibili schemi di flusso al verificarsi di una determinata condizione. Il blocco di controllo permette di scegliere quale schema di flusso debba essere eseguito, in funzione del valore di verità del controllo.





3. SCHEMA DI ITERAZIONE: itera l'esecuzione di un dato schema di flusso… azioni ripetute (*ciclo* o *loop*).



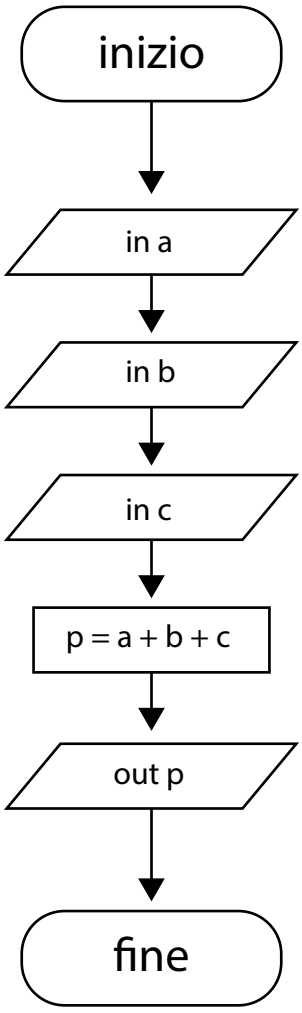
Ogni diagramma a blocchi non strutturato è sempre trasformabile in un diagramma a blocchi strutturato ad esso equivalente. "Teorema di Bohn-Jacopini"
Sottolineiamo che: due diagrammi sono equivalenti se, partendo dagli stessi dati iniziali, producono gli stessi risultati.

Facciamo qualche esempio.
Algoritmo sequenziale
Problema 1: *calcolare il perimetro di un triangolo.*

Sappiamo che per calcolare il perimetro di un triangolo ci basta sommare la misura dei tre lati.

$$p = a + b + c$$

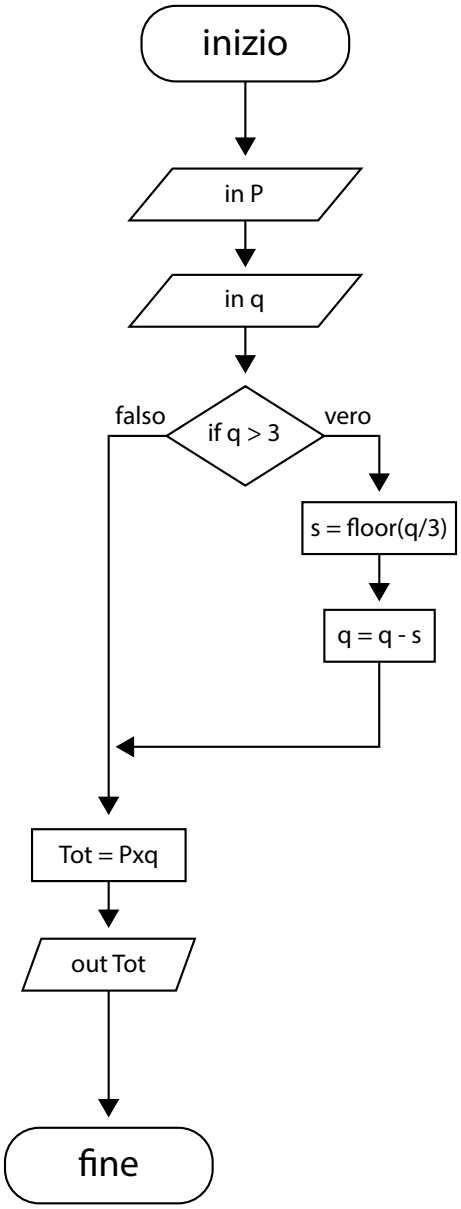
…per risolvere l'algoritmo, dovremo, dunque prendere input la lunghezza dei tre lati ed infine eseguire la somma.



Algoritmo di selezione

Problema 2: calcolare il prezzo da pagare per l'acquisto di una data quantità di prodotti acquistati considerando una promozione 3x2 cumulativa (ne prendo 3, ne pago 2; ne prendo 6, ne pago solo 4).

La soluzione è estremamente semplice: in input chiederò il prezzo P e la quantità q di prodotti che voglio acquistare. Se q è inferiore a 3 allora mi basterà moltiplicare qxP, caso contrario, calcolo la quantità s da sottrarre (divisione arrotondata per difetto) a q e, infine moltiplico la differenza (q-s) per il prezzo P.



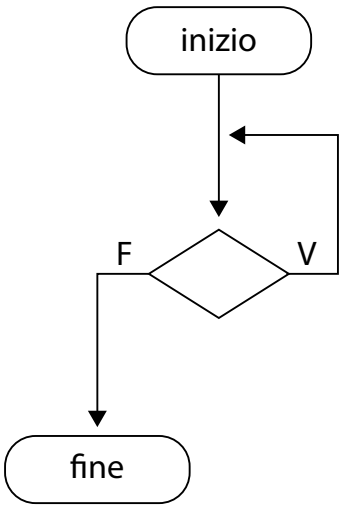
Algoritmo iterativo

Problema 3: stampare a schermo una successione di numeri che da 0, incrementando di +1, arrivi a 10.

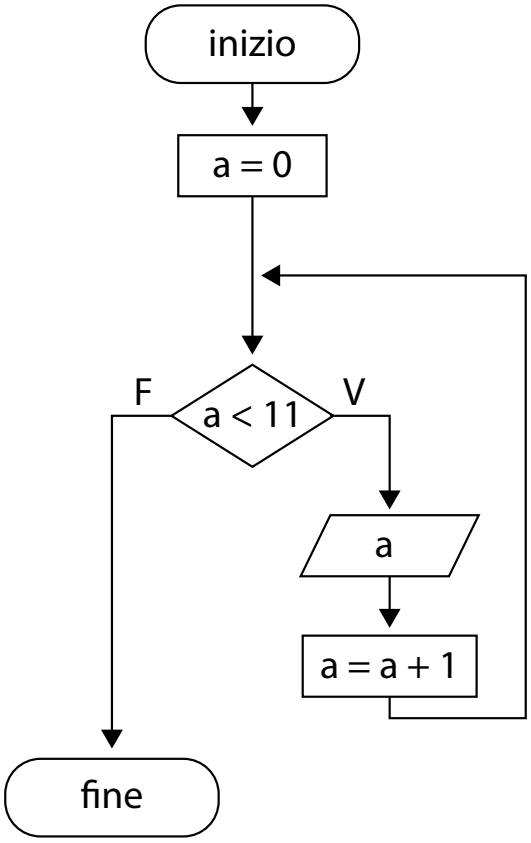
Fissato a 0 il valore della nostra variabile a, questo diventerà prima 1, poi, 2, poi 3, etc… fino ad a = 10.

Per poter eseguire una simile procedura, ciò di cui abbiamo bisogno è una struttura iterativa, nello specifico il ciclo *while*, una struttura che permette di ripetere una serie di istruzioni.

La forma per realizzare un ciclo *while* nei diagrammi di flusso è la seguente:



Inseriremo, allora, nel blocco di controllo la condizione a < 11 (fino a che a sarà inferiore a 11), allora fai qualcosa… semplicemente, incrementa a di 1 (a = a + 1).



Qualche appunto sugli algoritmi iterativi. Un ciclo si dice *definito* o *enumerativo*, quando è noto a priori quante volte deve essere eseguito; si dice *indefinito*, invece, quando non è possibile conoscere a priori il numero di volte esatto di esecuzione. Sia nel primo che nel secondo caso, un ciclo deve essere necessariamente eseguito un numero finito di volte!