

Dal problema al programma: le basi della programmazione

Codifichiamo gli algoritmi con i flow-chart

prof. Leonardo Essam Dei Rossi

ITT "M. Buonarroti" - Trento (TN)

Anno scolastico 2025/2026

- 1 I diagrammi a blocchi (o flow-chart)
- 2 Realizziamo i primi diagrammi a blocchi
 - Esempio #1: Area e perimetro di un quadrato
 - Esempio #2: Quanto costa andare a scuola?
- 3 Le variabili e le costanti
 - Le variabili
 - Esempio #3: Secchi con l'acqua
 - Le costanti
- 4 Esercizi

- 1 I diagrammi a blocchi (o flow-chart)
- 2 Realizziamo i primi diagrammi a blocchi
 - Esempio #1: Area e perimetro di un quadrato
 - Esempio #2: Quanto costa andare a scuola?
- 3 Le variabili e le costanti
 - Le variabili
 - Esempio #3: Secchi con l'acqua
 - Le costanti
- 4 Esercizi

I diagrammi a blocchi (o flow-chart) (1)

Esistono diversi modi per scrivere un algoritmo. il più semplice, e anche il più diffuso, è quello che utilizza i diagrammi a blocchi (o *flow-chart*, letteralmente tradotto con *diagrammi di flusso*): si tratta di una rappresentazione grafica, di facile interpretazione per chiunque e molto efficace perché permette di individuare “a colpo d’occhio” la struttura dell’algoritmo.

Vengono utilizzati simboli grafici (chiamati *blocchi*) che contengono le istruzioni: i blocchi hanno quattro forme geometriche diverse per distinguere le quattro tipologie possibili di operazioni che permettono di rappresentare tutti gli algoritmi.

I diagrammi a blocchi (o flow-chart) (2)

Le quattro tipologie di blocchi sono:

- **Blocco di inizio/fine:** è un blocco di forma ovale che viene utilizzato solo come primo e ultimo blocco del diagramma;

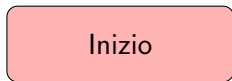


Figura 1.1: Blocco di inizio

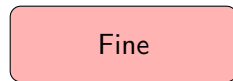


Figura 1.2: Blocco di fine

I diagrammi a blocchi (o flow-chart) (3)

- **Blocco di trasferimento di informazione (comunicazione I/O):** è un blocco a forma di parallelogramma che viene utilizzato dare/ricevere informazioni all'/dall'utente, ovvero per eseguire operazioni di INPUT e OUTPUT;

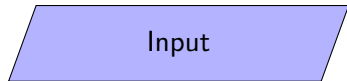


Figura 1.3: Blocco di I/O

I diagrammi a blocchi (o flow-chart) (4)

- **Blocco di elaborazione:** è un blocco di forma rettangolare, che viene utilizzato per contenere istruzioni che effettuano trasformazione di dati;

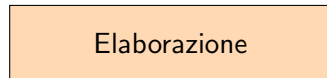


Figura 1.4: Blocco di elaborazione

I diagrammi a blocchi (o flow-chart) (5)

- **Blocco di decisione (condizionale):** è un blocco di forma romboidale, al suo interno vengono effettuate operazioni di confronto (test) che si riassumono in domande che possono avere come risultato esclusivamente due valori: SÌ oppure NO (cioè VERO oppure FALSO);

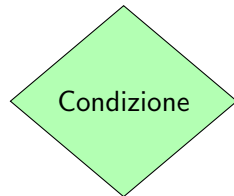


Figura 1.5: Blocco di decisione (condizionale)

I diagrammi a blocchi (o flow-chart) (6)

Il **blocco di decisione (condizionale)** è un blocco particolare perché permette di modificare la sequenza e l'ordine di esecuzione delle istruzioni del programma.

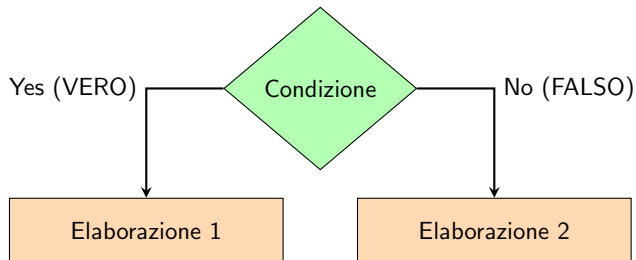


Figura 1.6: Blocco di decisione (condizionale) ramificato

I diagrammi a blocchi (o flow-chart) (7)

In ogni diagramma deve essere presente un solo blocco di inizio (Figura 1.1) e, soprattutto, un solo blocco di terminazione (Figura 1.2): anche se nel diagramma saranno presenti percorsi alternativi proprio in base ai risultati della istruzione di decisione, tutti i percorsi alternativi devono ricongiungersi prima della fine del diagramma.

- 1 I diagrammi a blocchi (o flow-chart)
- 2 Realizziamo i primi diagrammi a blocchi
 - Esempio #1: Area e perimetro di un quadrato
 - Esempio #2: Quanto costa andare a scuola?
- 3 Le variabili e le costanti
 - Le variabili
 - Esempio #3: Secchi con l'acqua
 - Le costanti
- 4 Esercizi

Realizziamo i primi diagrammi a blocchi

Ricordiamo dalla lezione precedente [1] i seguenti concetti:

- **Affrontare un problema:** per individuare la soluzione potrebbe essere necessario cercare di comprendere meglio il problema, cioè effettuare un'analisi approfondita della situazione e, nei casi più complessi, individuare una strategia risolutiva;
- **Ricerca la strategia risolutiva:** ovvero la modalità con la quale si risolve il problema, cioè l'idea con la quale il programmatore [...] trova la “chiave di soluzione” del problema;
- **Scomporre in istruzioni/passi:** cioè descrivere l'insieme delle operazioni (passi) che devono essere effettuati riportandoli nell'ordine in cui devono essere eseguiti (sequenza di operazioni).

Esempio #1: Area e perimetro di un quadrato (1)

Per prima cosa, definiamo la formula per calcolare il perimetro e l'area di un quadrato:

$$P = l \cdot 4 \quad A = l \cdot l$$

Realizziamo un diagramma a blocchi che descrive le operazioni necessarie per effettuare il calcolo del perimetro e dell'area di un quadrato: per prima cosa descriviamo l'algoritmo in pseudocodice e successivamente lo trasformiamo in flow-chart.

Tratto da: [2, p. 332]

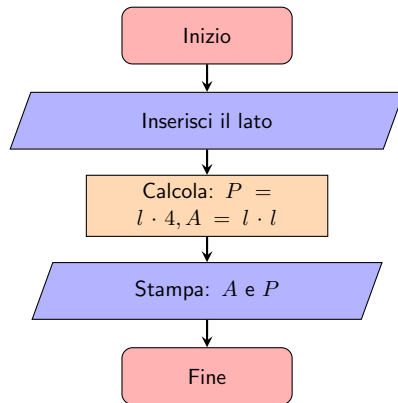
Esempio #1: Area e perimetro di un quadrato (2)

Scriviamo l'algoritmo in pseudocodifica:

- 1 Inserisci il valore del lato;
- 2 Calcola il perimetro moltiplicando il lato per 4;
- 3 Calcola l'area moltiplicando il lato per il lato;
- 4 Visualizza il valore del perimetro e dell'area.

Esempio #1: Area e perimetro di un quadrato (3)

Scriviamo ora il flow-chart
dell'algoritmo appena descritto:



Esempio #2: Quanto costa andare a scuola? (1)

Vediamo un secondo esempio.

Quanto costa andare a scuola?

Calcoliamo il costo settimanale di carburante utilizzato per andare a scuola.

Innanzitutto individuiamo le operazioni da eseguire, cioè l'algoritmo che risolve il problema:

- 1 Calcolare il numero di litri di benzina consumati;
- 2 Calcolare il costo totale del carburante giornaliero;
- 3 Determinare il costo settimanale.

Esempio #2: Quanto costa andare a scuola? (2)

A questo punto è necessario avere a disposizione alcuni dati da fornire al programma, in modo che esso possa elaborarli.

Questi dati, chiamati dati di ingresso (*input*), sono:

- 1 Il costo di un litro di carburante;
- 2 Il numero di chilometri tra la casa e la scuola;
- 3 Il numero di chilometri che il mezzo utilizzato effettua con un litro di carburante.

Esempio #2: Quanto costa andare a scuola? (3)

Definizione: output di un programma

I risultati dell'elaborazione prendono il nome di dati in uscita (o output) del programma e vengono comunicati all'utente.

In questo semplice esempio possiamo osservare come viene evidenziata la struttura generale di ogni algoritmo che possiamo schematizzare in tre fasi fondamentali: input dai dati iniziali, elaborazione mediante il programma e output dei risultati.



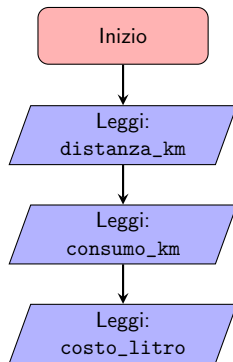
Esempio #2: Quanto costa andare a scuola? (4)

Una possibile codifica in pseudocodice è la seguente:

- 1 Inserimento della distanza in km;
- 2 Inserimento del consumo in km;
- 3 Inserimento del costo del carburante al litro;
- 4 Calcolo del numero di litri consumati;
- 5 Calcolo del costo giornaliero;
- 6 Calcolo del costo settimanale;
- 7 Visualizzazione dei risultati.

Esempio #2: Quanto costa andare a scuola? (5)

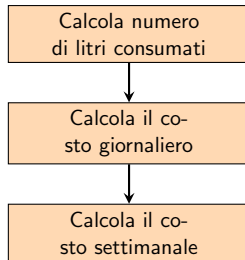
Parte di flow-chart per la parte di input (dall'utente):



(continua)

Esempio #2: Quanto costa andare a scuola? (6)

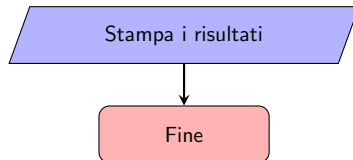
Parte di flow-chart per la parte di elaborazione:



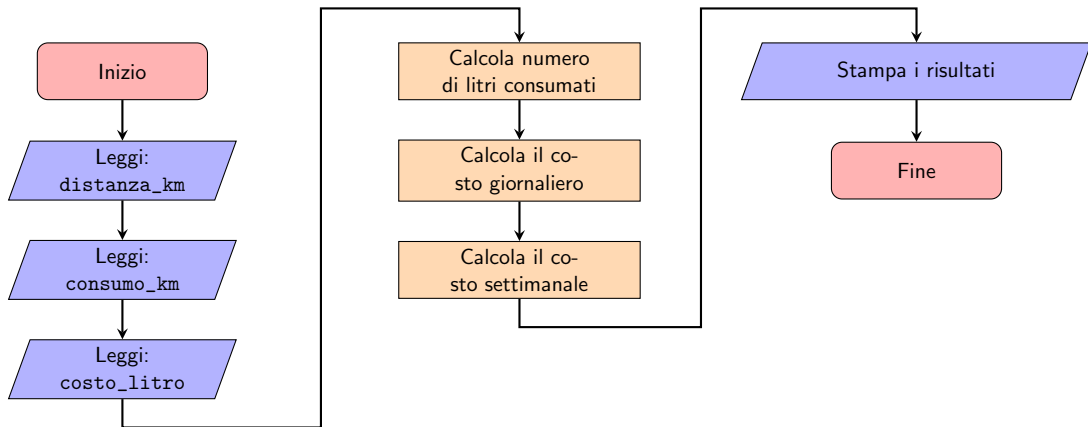
(continua)

Esempio #2: Quanto costa andare a scuola? (7)

Parte di flow-chart per la parte di output (stampa a schermo):



Esempio #2: Quanto costa andare a scuola? (8)



Esempio #2: Quanto costa andare a scuola? (9)

Su questo esempio possiamo fare alcune riflessioni:

- L'algoritmo che abbiamo scritto ha la proprietà di essere generico, ovvero che può essere utilizzato per la risoluzione di problemi simili senza che vengano specificati il tipo di automezzo utilizzato, il tipo di percorso effettuato o di carburante impiegato;
- La sequenza delle istruzioni è ordinata, cioè esiste un ordine preciso in base al quale vengono eseguite le istruzioni, che non possono essere scambiate di posto tra loro;

Esempio #2: Quanto costa andare a scuola? (10)

- Il numero di operazioni è finito: ciò significa che le istruzioni possono essere anche moltissime, ma non in numero infinito, altrimenti l'algoritmo non terminerebbe mai;
- Ogni operazione, inoltre, deve essere chiara, in modo che l'esecutore possa comprendere ed eseguire ogni istruzione senza errori di interpretazione;
- Come precedentemente detto, è presente un solo inizio e una fine, sia che venga prodotto un risultato valido sia che non esista soluzione;

Esempio #2: Quanto costa andare a scuola? (11)

- I risultati generati devono essere gli stessi tutte le volte che l'algoritmo viene eseguito sugli stessi dati (determinismo), anche se dovesse essere usato un elaboratore diverso.

Definizione: algoritmo deterministico

Un algoritmo deterministico è un algoritmo dove, a parità di input, ogni passo dell'esecuzione è univocamente determinato e produce sempre la stessa sequenza di stati¹ fino allo stesso output finale.

¹Lo stato è la configurazione del programma al tempo t di esecuzione.

- 1 I diagrammi a blocchi (o flow-chart)
- 2 Realizziamo i primi diagrammi a blocchi
 - Esempio #1: Area e perimetro di un quadrato
 - Esempio #2: Quanto costa andare a scuola?
- 3 Le variabili e le costanti
 - Le variabili
 - Esempio #3: Secchi con l'acqua
 - Le costanti
- 4 Esercizi

Le variabili (1)

Nell'esempio precedente abbiamo indicato tra le operazioni il blocco che permette di effettuare la comunicazione con l'utente, cioè di descrivere le istruzioni con le quali avviene lo scambio di dati (in questo caso di numeri) che sono necessari per produrre i risultati.

In tutti i linguaggi di programmazione particolare attenzione è dedicata alle operazioni che permettono di memorizzare le informazioni in quanto i dati sono la *"linfa vitale"* dell'algoritmo, poiché senza dati non può esistere l'elaborazione.

Le variabili (2)

Vengono messe a disposizione del programmatore le istruzioni necessarie per definire i “*contenitori dei dati*” e per effettuare operazioni su di essi: questi “*contenitori*”, chiamati variabili. Sono praticamente delle aree (di memoria) dove vengono “depositati” i dati prima (e dopo) le elaborazioni. Il programmatore dà a ciascuna di esse un nome (identificatore) in modo da poterle distinguere e poter in seguito fare riferimento a esse nelle istruzioni di elaborazione.

Q: *quale memoria?*

A: *la memoria RAM.*

Le variabili (3)

Possiamo pensare alle variabili come a dei “*cassetti contenitori di dati*” posti uno vicino all'altro sui quali “*attacciamo un'etichetta*” con un nome, di fantasia oppure secondo il “*buon senso*”, per ricordarci il loro contenuto e per distinguerli tra loro.

Esempio

Se dobbiamo memorizzare il peso e l'altezza di un alunno è preferibile dare alle variabili i nomi nome, peso e altezza piuttosto che pippo e paperino oppure A e B, così che direttamente dal nome della variabile (identificatore) possiamo immediatamente ricordarci cosa contiene, soprattutto se guardiamo il programma dopo giorni o settimane!

Naturalmente, tutti i nomi che attribuiamo alle variabili devono essere differenti per poterle individuare univocamente!

Esempio #3: Secchi con l'acqua (1)

Si vuole descrivere l'algoritmo che permetta di versare 2 litri in un secchio della capacità di 4 litri utilizzando solo due secchi con capacità volumetrica rispettivamente di 4 e 3 litri.

In questo esempio “abbiamo proprio bisogno” di due contenitori: sfruttiamo due variabili nelle quali inseriremo il numero dei litri presenti in ciascun secchio.

Come identificatore per queste variabili possiamo scegliere ad esempio `secchio3` e `secchio4`, dove naturalmente con `secchio3` indichiamo il secchio in grado di contenere 3 litri con e `secchio4` il secondo secchio con capacità di 4 litri.

Esempio #3: Secchi con l'acqua (2)

A differenza degli esempi precedenti, dove le operazioni da risolvere erano immediatamente individuate, in questo problema il procedimento è un po' più complesso in quanto è necessario individuare una strategia risolutiva.

Osserviamo che la differenza tra i due contenitori è di 1 litro di capacità, questa quantità ci può servire per ottenere la quantità di 2 litri desiderata:

- 1 Riempio per primo il secchio da 4 litri;
- 2 Lo travaso in quello da 3 litri in modo che rimanga 1 litro nel primo secchio;
- 3 Ora svuoto il secchio da 3 litri;
- 4 Travaso il litro presente nel secchio da 4 litri in quello da 3 litri;

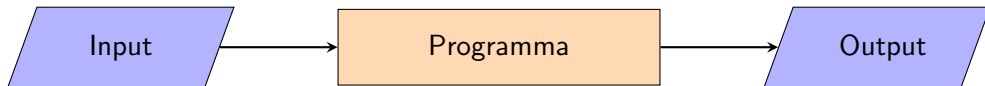
Esempio #3: Secchi con l'acqua (3)

- 5 Riempio nuovamente il secchio da 4 litri;
- 6 Travaso due litri in quello da 3 litri, dove 1 litro è già presente;
- 7 Nel secchio da 4 litri sono rimasti 2 litri.

Questo programma non ha né dati in input né in output ma semplicemente una sequenza di istruzioni di elaborazione in quanto i dati sono intrinsecamente indicati dal testo.

Esempio #3: Secchi con l'acqua (4)

Ricordiamo la figura:



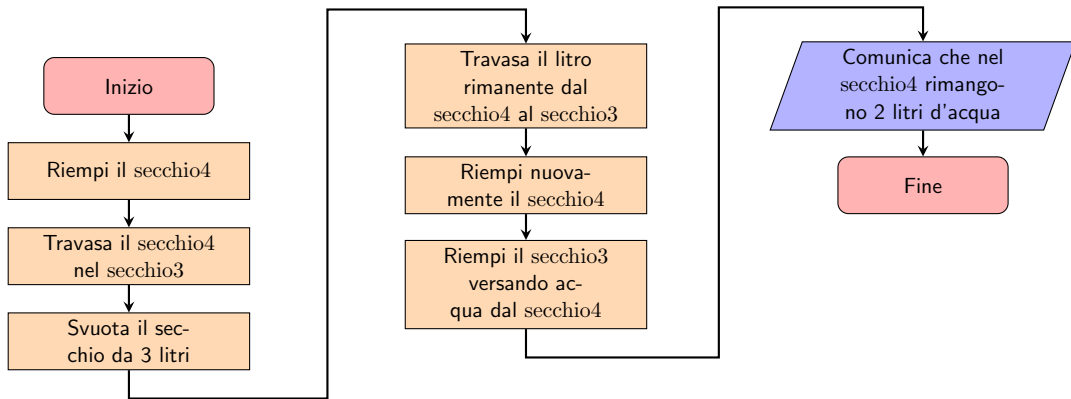
Sulla base di quanto mostrato in figura, quale incongruenza si presenta rispetto all'algoritmo illustrato in precedenza?

Esempio #3: Secchi con l'acqua (5)

Non è proprio vero che non ci sono i dati di input: in questo caso i dati di input, che sono i dati di partenza sui quali il nostro programma dovrà compiere le operazioni, sono presenti all'interno del testo del problema stesso, fanno cioè parte delle specifiche del problema.

Esempio #3: Secchi con l'acqua (6)

Scriviamo ora il flow-chart dell'algoritmo appena descritto:



Le costanti (1)

Oltre alle necessità di memorizzare i dati che vengono elaborati nel corso dell'esecuzione del programma esistono anche delle situazioni in cui alcune variabili contengono dati che “non devono essere variati” e si mantengono uguali per tutta l'esecuzione del programma: questi dati prendono il nome di costanti e nei linguaggi di programmazione è sempre presente la possibilità di distinguerle dalle variabili, in modo che il programmatore non le modifichi, neppure per errore.

Le costanti (2)

Ad esempio tra le costanti abbiamo quelle dette “universalì”, cioè che non cambiano il loro valore neppure in programmi differenti, come la velocità della luce, il valore di pi-greco (π), il numero dei mesi dell’anno, le ore del giorno ecc. oppure costanti temporanee che valgono solo per periodi di tempo limitato, come il valore del cambio delle monete, il costo dei beni in un supermercato, il prezzo del biglietto del cinema, il numero degli alunni di una classe ecc.

Solitamente nei diagrammi di flusso non vengono indicate differenze tra variabili e costanti.

- 1 I diagrammi a blocchi (o flow-chart)
- 2 Realizziamo i primi diagrammi a blocchi
 - Esempio #1: Area e perimetro di un quadrato
 - Esempio #2: Quanto costa andare a scuola?
- 3 Le variabili e le costanti
 - Le variabili
 - Esempio #3: Secchi con l'acqua
 - Le costanti
- 4 Esercizi

Per ciascuna delle seguenti situazioni si descriva l'algoritmo risolutivo sia in pseudocodifica che mediante flow-chart:

- 1 Si descriva l'algoritmo che elenca le operazioni necessarie per spedire una email allegando una fotografia;
- 2 Si descriva l'algoritmo che elenca le operazioni del mattino, dalla sveglia all'uscita di casa per andare a scuola;
- 3 Si descriva l'algoritmo che permette di ottenere in un contenitore 2 litri utilizzando tre contenitori rispettivamente di volume 3 litri, 6 litri e 10 litri;
- 4 Si descriva l'algoritmo che permette di ottenere in un contenitore 5 litri utilizzando tre contenitori rispettivamente di volume 4 litri, 7 litri e 13 litri.

- [1] L. E. Dei Rossi, *Corso di Laboratorio di Tecnologie Informatiche*, 1, 2025.
- [2] P. Camagni e R. Nikolassy, *TeknoLab: Tecnologie informatiche E Laboratorio: Office automation, Sviluppo del Pensiero computazionale, scratch e applicazioni per arduino, app inventor per android, Algobuild e Codifica C++*. Hoepli, 2017.