

Introduzione alla programmazione

Risolvere un problema

- Per risolvere un problema si procede innanzitutto all'individuazione
 - Delle informazioni, dei dati noti
 - Dei risultati desiderati
- Il secondo passo consiste nell'individuazione di un procedimento adeguato a risolvere il problema
- Si scompone infine il procedimento con l'individuazione delle operazioni necessarie per il calcolo di risultati intermedi

Dal problema al programma

- Per realizzare un programma che risolva un problema è necessario quindi:
 - Individuare i dati in ingresso e in uscita
 - Individuare un procedimento adeguato
 - Scomporre il procedimento in una sequenza di azioni elementari ed univoche

Algoritmi

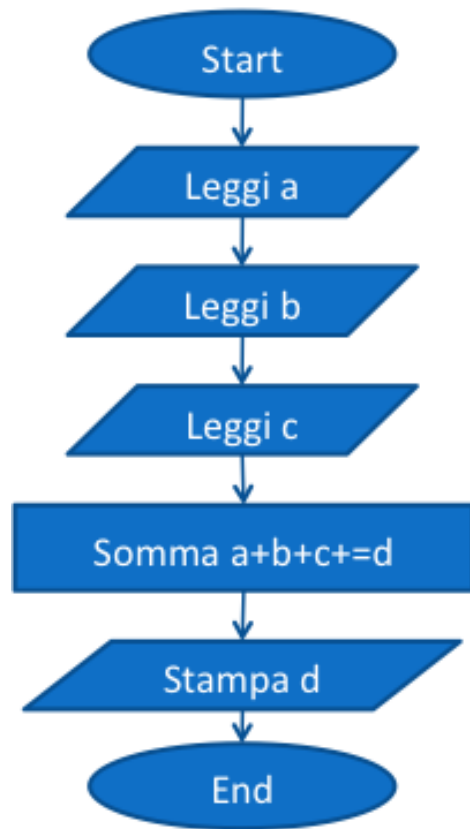
- Per indicare la scomposizione del problema viene spesso utilizzato il termine algoritmo:
“Insieme finito di istruzioni elementari univocamente interpretabili che, eseguite in un ordine stabilito, forniscono la soluzione di un problema in un numero finito di passi”
- I flow-chart sono un “linguaggio” utile per rappresentare e descrivere gli algoritmi

Flow-chart

- Il flusso di esecuzione può essere rappresentato graficamente con un diagramma di flusso (flow-chart):
 - Elimina le ambiguità utilizzando una logica di selezione ambivalente e linee di flusso univoche
 - Si basa su pochi simboli
 - E' un linguaggio universale

Esempio di Flow-chart

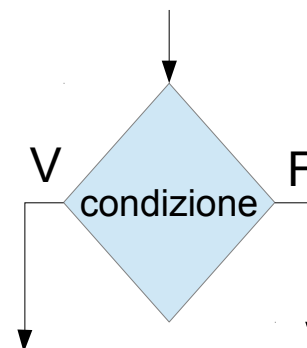
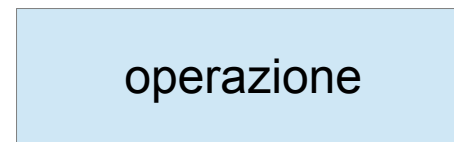
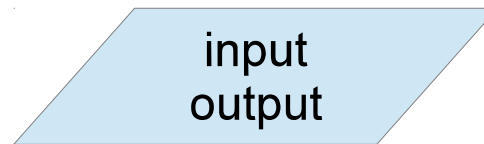
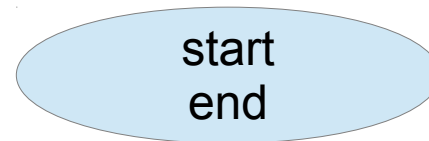
Diagramma di flusso



Dati tre numeri a, b, c
stampare la somma

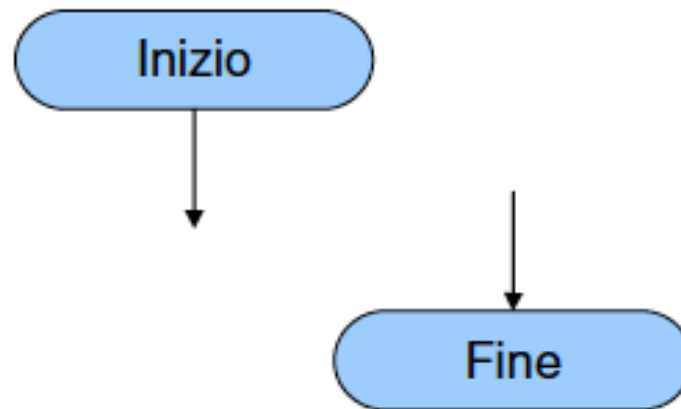
Flow- chart: i simboli

- **Ovale** inizio e fine diagramma
- **Parallelogrammo**, indica dati in input e output
- **Rettangolo** indica operazione di calcolo
- **Rombo** indica la scelta (vero/falso) al verificarsi di una condizione



Flow-chart

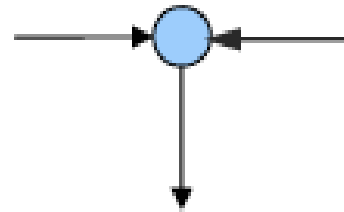
- Innanzitutto un programma, e di conseguenza una sua rappresentazione grafica, deve sempre avere un inizio ed una fine



- Tra l'inizio e la fine vi deve sempre essere almeno una istruzione

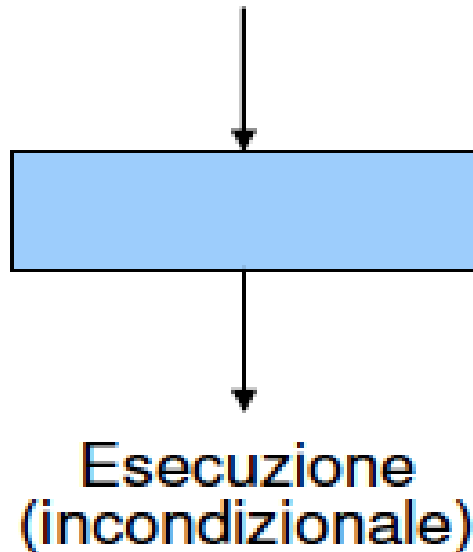
Flow-chart

- La risoluzione di un problema consiste nell'esecuzione ordinata di una sequenza di operazioni
 - L'ordine nell'esecuzione delle istruzioni è fondamentale
 - Nei flow-chart è garantito dall'orientamento delle frecce che collegano i blocchi



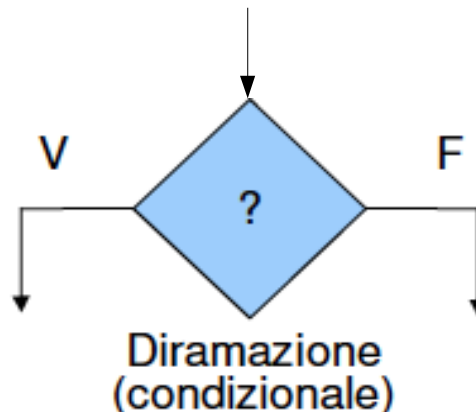
Flow-chart

- A volte le azioni vanno eseguite “sempre”. In tal caso diciamo che si tratta di operazioni non condizionali



Flow-chart

- Possono essere presenti istruzioni condizionali, la cui esecuzione dipende cioè da scelte effettuate in base ai dati
 - Concettualmente, possiamo immaginare che il flusso di esecuzione si ramifichi (in base ad una condizione decido se eseguire un'operazione OPPURE un'altra)

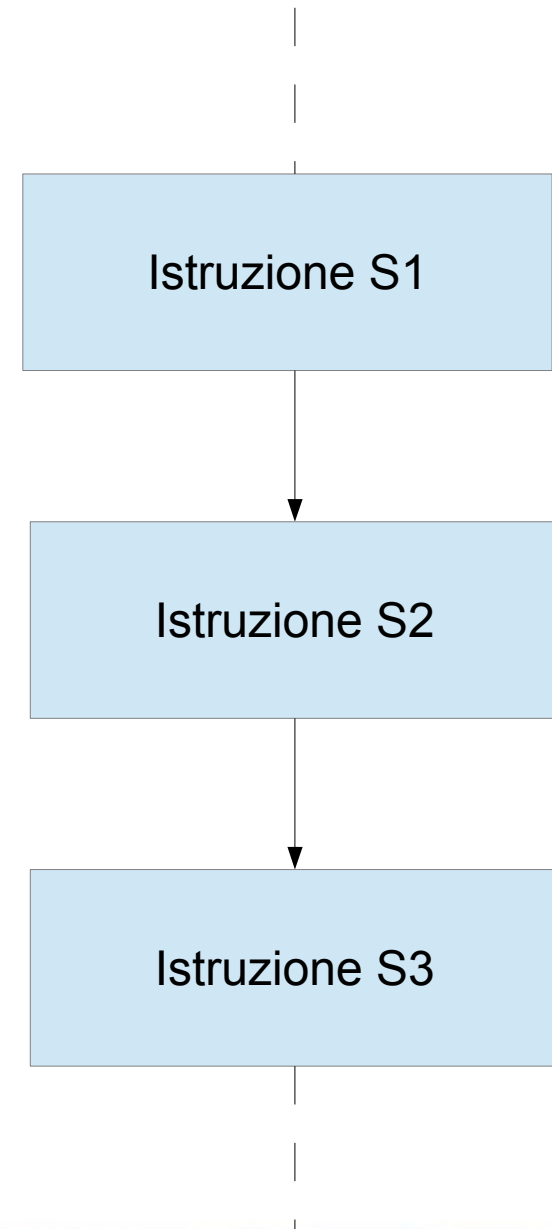


Costrutti di programmazione

- Esistono alcuni costrutti di base tipici dei linguaggi di alto livello di programmazione
- Qualsiasi algoritmo rappresentabile con un flow-chart strutturato può essere tradotto in un programma utilizzando appena tre strutture:
 - Sequenza
 - Selezione
 - Iterazione

Sequenza

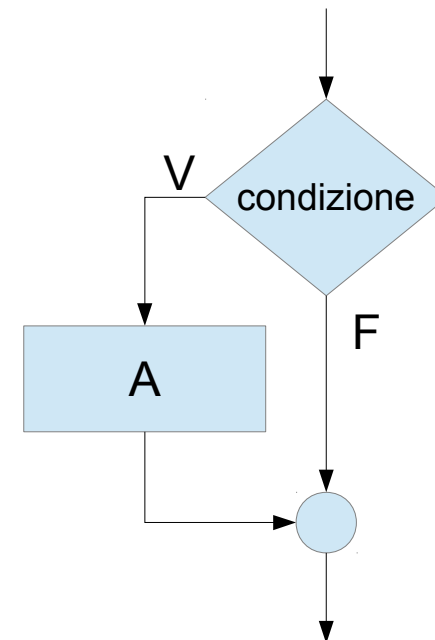
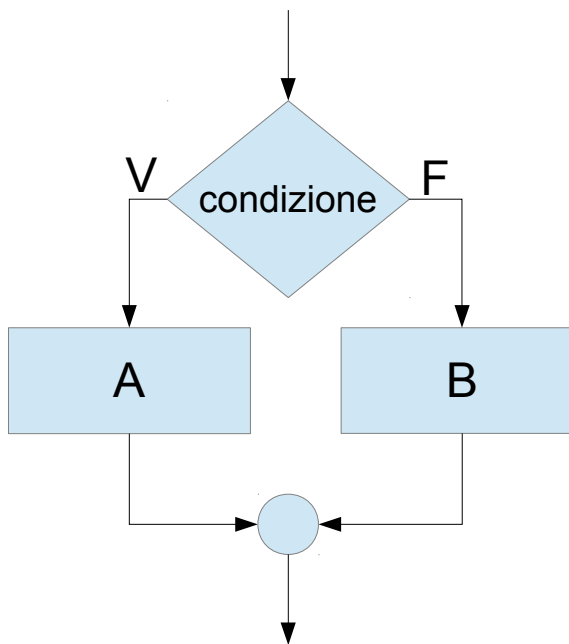
- Le istruzioni sono una di seguito all'altra



Selezione

- È un costrutto tipico che si usa per scegliere tra due alternative
- Il blocco else è opzionale

```
if (condizione)  
then  
fare A  
[else  
    fare B  
end]
```



Iterazione

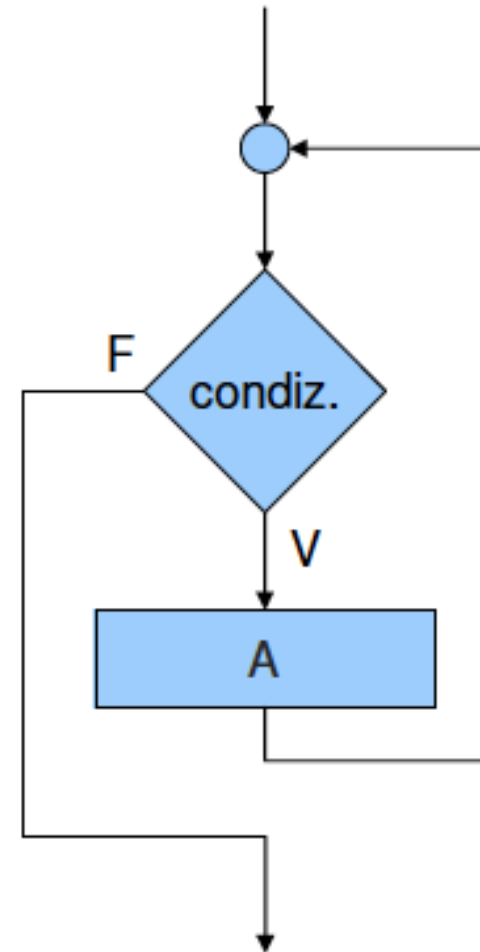
- Per poter ripetere lo stesso gruppo di istruzioni più volte, è necessario utilizzare un altro costrutto: il ciclo (o loop)
- E' necessario definire una condizione per stabilire quante volte ripetere il blocco di istruzioni

Iterazione

- Un blocco può essere eseguito “fintanto che” una certa condizione è verificata ed interrompere l’esecuzione non appena questa diventa falsa
- Se allo stato iniziale la condizione è falsa, le istruzioni non vengono mai eseguite, questo è detto ciclo ad ingresso controllato

Iterazione: while

- Il costrutto while realizza un ciclo ad ingresso controllato
`while (condizione)`
`fare A`



Iterazione

- Un blocco può essere eseguito anche “fino a quando” non si verifica una condizione prestabilita: questo è detto ciclo ad uscita controllata
- In questo caso prima viene eseguita un'iterazione e poi controllato il valore di verità della condizione

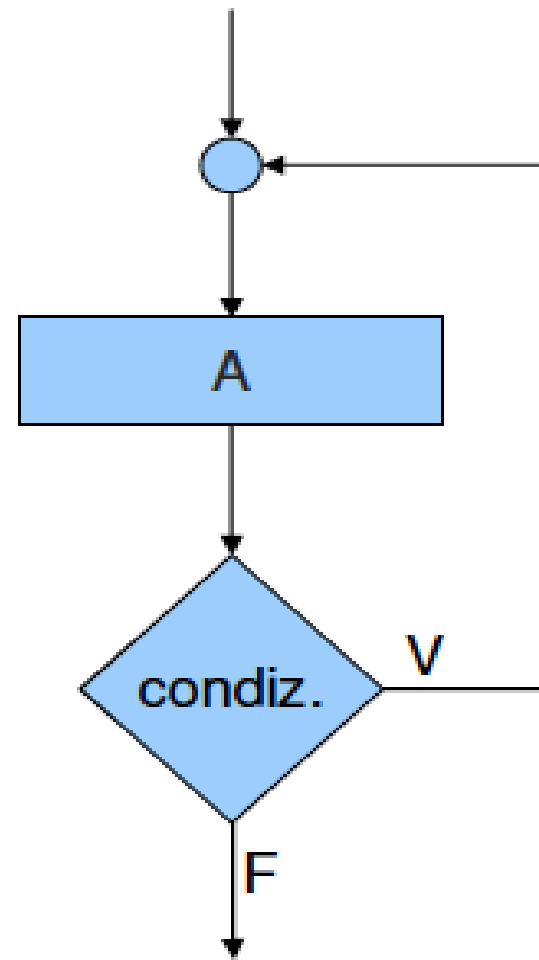
Iterazione: do while

- Il costrutto do-while (o repeat-until) realizza un ciclo ad uscita controllata

do

fare A

while (condizione)

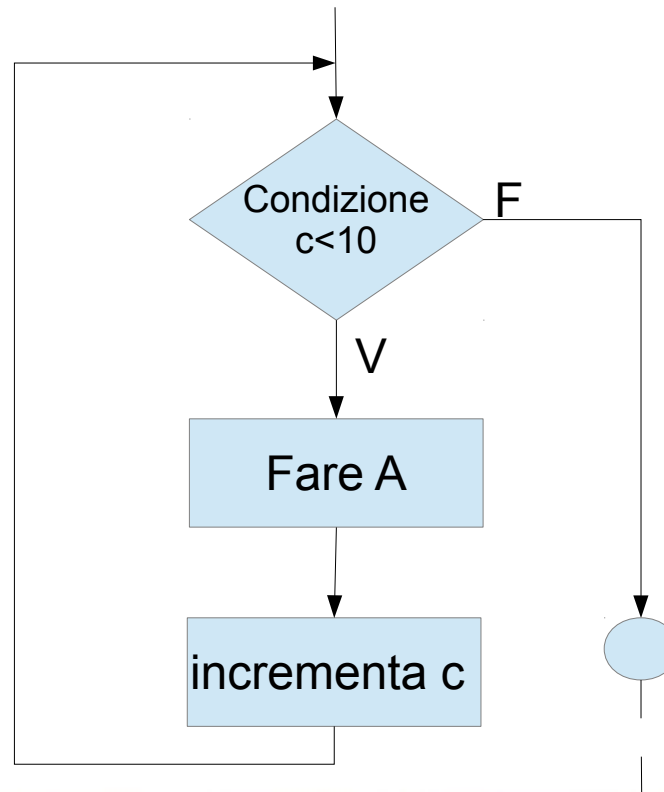


Iterazione: for

- Un terzo costrutto viene impiegato se è noto il numero di iterazioni da eseguire

```
for (Val_iniziale; Cond; Incremento)
```

```
fare A;
```



Linguaggi di programmazione

- Un linguaggio di programmazione è un linguaggio formale costituito da simboli (alfabeto), combinati in sequenze in base a regole precise (sintassi) e a cui viene associato un determinato significato (semantica)
- Le famiglie di linguaggi
 - Linguaggi assembler
 - Linguaggi funzionali
 - Linguaggi imperativi (BASIC)
 - Linguaggi a oggetti (Java) etc

Linguaggi di programmazione

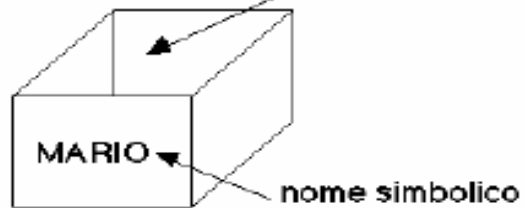
- Un calcolatore elettronico (PC) utilizza esclusivamente l'alfabeto binario
 - I linguaggi consentono quindi di descrivere le operazioni da far compiere al calcolatore in maniera comprensibile all'uomo (digitando istruzioni scritte ad esempio in linguaggio Java) che verranno poi tradotte in binario (01010101)
- E' necessaria un'operazione di traduzione dal linguaggio di programmazione al codice comprensibile dal calcolatore, effettuata per mezzo di opportuni programmi "traduttori"
- Questi traduttori si dividono in due categorie: compilatori e interpreti

Le informazioni nel calcolatore

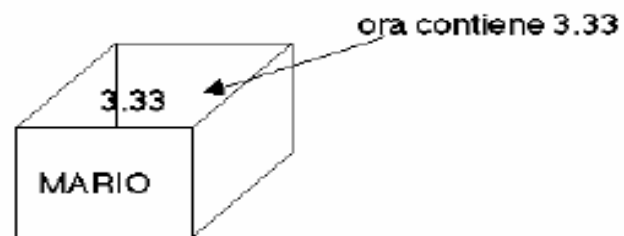
- I calcolatori utilizzano esclusivamente numeri, mentre il linguaggio (ed il cervello umano) gestisce più facilmente i nomi
- I linguaggi di programmazione permettono quindi di fare riferimento ai dati utilizzando dei nomi, detti variabili
- Una variabile può essere vista come un contenitore avente un nome ed un contenuto
 - Il nome variabile vuole enfatizzare il fatto che il contenuto può cambiare.

Le variabili

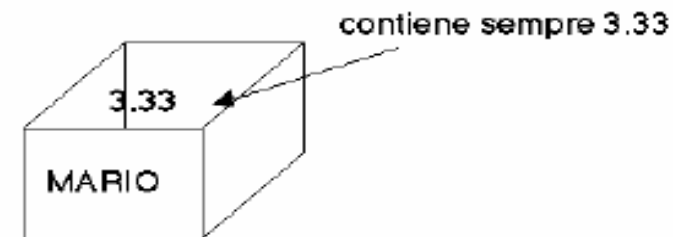
DICHIARO UNA VARIABILE: inizialmente vuota



CAMBIO IL CONTENUTO DI MARIO (scrivo MARIO):



ESAMINO IL CONTENUTO DI MARIO (leggo MARIO):



il contenuto è 3.33

Tipi di dato

- Il contenuto di una variabile all'interno del calcolatore è una sequenza di bit
 - occorre determinare quale valore contiene la variabile, cioè come interpretare tali bit.
- I linguaggi di programmazione associano ad ogni variabile informazioni anche sul tipo di valore che essa contiene.
- L'attribuzione di un significato particolare alla sequenza binaria è fatto attraverso il concetto di tipo di dato

Tipi di dato

- Un insieme minimo di tipi di dato che ci interessa risulta composto da:
 - numeri interi
 - numeri in virgola mobile
 - caratteri
 - stringa che può essere vista come una sequenza di caratteri
- In genere sono disponibili strumenti che consentono di convertire un dato da un tipo ad un altro

Vettori

- Una variabile contiene solo un dato mentre molti algoritmi gestiscono una serie di dati ordinati e dello stesso tipo.
- In questo caso si utilizzano i vettori, un insieme ordinato di variabili dello stesso tipo
- Per accedere ad un elemento specifico se ne indica la posizione
- Esempio un vettore a di interi:
 $a = [1, 4, 33, 12, 42, 66]$