

FONDAMENTI DI PROGRAMMAZIONE C / C++

Docente: Armando Valentino

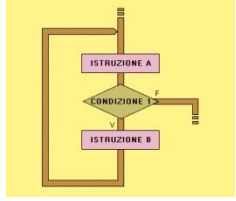
In collaborazione con:



per una crescita intelligente,
sostenibile ed inclusiva

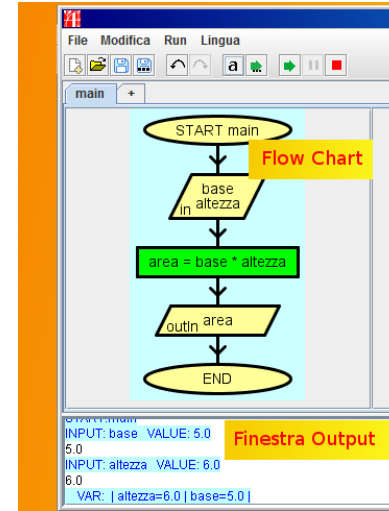
www.regione.piemonte.it/europa2020

INIZIATIVA CO-FINANZIATA CON FSE



ALGORITMI

PROGRAMMAZIONE STRUTTURATA



Prof. Armando Valentino

Definizione di Algoritmo

- Un algoritmo è sequenza finita di azioni elementari che descrivono la soluzione di un problema in modo completo
- Ogni algoritmo è un insieme **finito di azioni** e deve **terminare** dopo un numero finito di istruzioni.
- **Completo:** deve considerare tutti i casi possibili che si possono verificare durante l'esecuzione e per ogni caso può indicare la soluzione da seguire.

CARATTERISTICHE DI UN ALGORITMO

- Composto da Istruzioni che l'esecutore è in grado di eseguire
- Termina in un tempo finito
- Produce risultati che si possono descrivere
- **Deterministico:** stessi input producono stessi output
- **Generale:** deve risolvere una serie di problemi dello stesso tipo, non un singolo caso particolare.

DIFFERENZA TRA ALGORITMO E PROGRAMMA

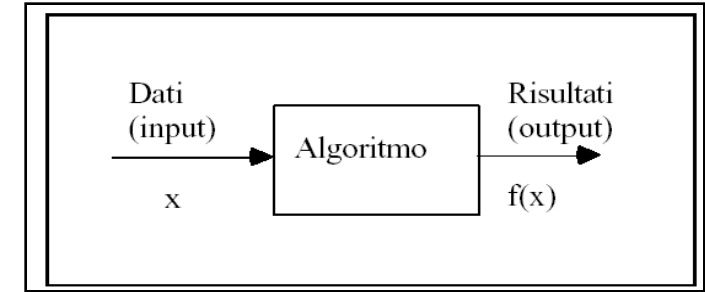
Un **ALGORITMO** codificato in un linguaggio di programmazione viene detto **PROGRAMMA**

I dati dell'algoritmo

I dati permettono la comunicazione tra utente e programma

I dati di input devono essere forniti dall'utente al programma

I dati di output sono il risultato dell'elaborazione, che **vengono restituiti dal programma all'utente**



I dati si possono essere:

COSTANTI

sono dati che rimangono sempre uguali, non variano

VARIABILI

sono dati che assumono valori diversi durante l'esecuzione del

programma

Tipi di dati:

Numerici (interi o reali)

Alfanumerici

Stringhe

Booleano (valori vero o falso)

- **Dati di INPUT:** sono quelli che devono essere forniti dall'esterno per poter risolvere il problema.
 - **Dati di OUTPUT:** sono quelli che vengono comunicati all'esterno come risultato della soluzione del problema.
 - **Dati INTERNI o di LAVORO:** sono i dati utilizzati nella trasformazione compiuta dall'algoritmo ma trasparenti all'utente (non sono forniti in output).
-
- **Dati NUMERICI:** dati che contengono numeri e sui quali si possono effettuare operazioni aritmetiche. Possono essere ulteriormente suddivisi in:
 - **INTERI:** dati numerici che non prevedono cifre decimali
 - **REALI:** dati numerici che prevedono cifre decimali
 - **Dati ALFANUMERICI (o STRINGHE):** sono i dati che contengono caratteri alfabetici (A,B,...), caratteri speciali (\$,%,&,...) e cifre (0,1,2,...) sulle quali non sono possibili operazioni aritmetiche (ad esempio il codice fiscale).

Dominio dei dati e vincoli

All'interno di un algoritmo, **non tutti i valori di un tipo di dati sono utilizzati, ma solo un intervallo.**

Esempio:

- la media dei voti di uno studente della scuola è tra 1 e 10
- La media dei voti di uno studente universitario è tra 1 e 30

L'insieme dei valori ammissibili dei dati si dice **dominio**.

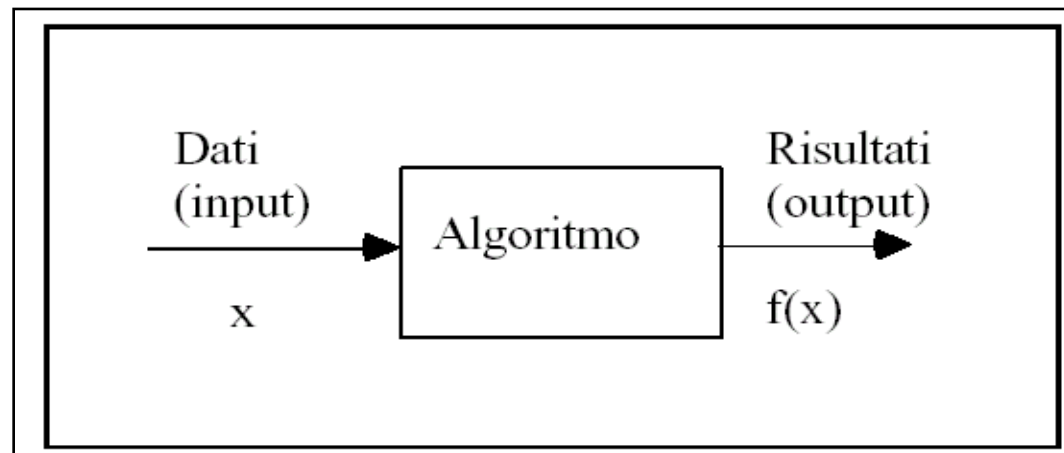
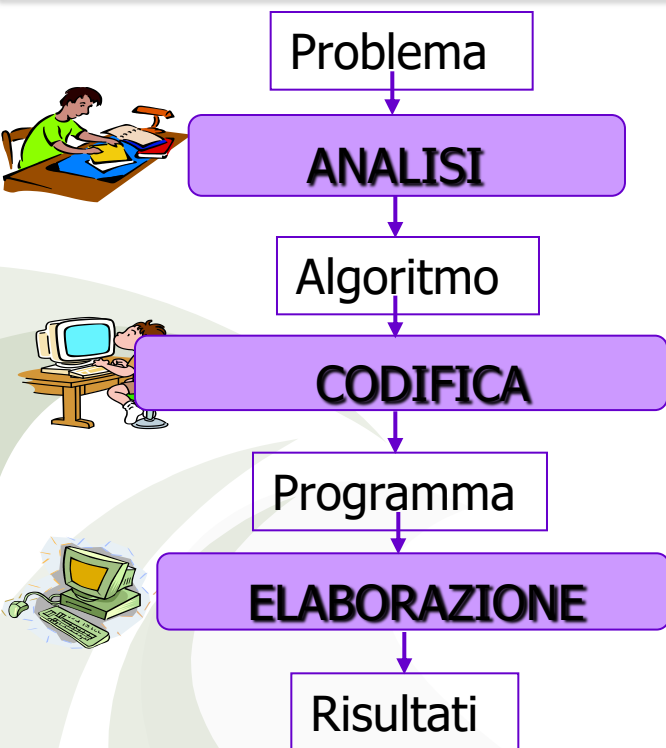
Il **dominio** può essere un intervallo dei valori (**subrange**) o un elenco di valori (**enumerazione**)

A volte si devono inserire dei vincoli ulteriori.

Esempio: il numero massimo dei giorni dipende dal mese considerato

IMPORTANTE!!: Stabilire il **dominio** e i **vincoli** serve per **effettuare i controlli sull'inserimento dei dati** e sui valori che i dati assumono durante l'elaborazione

Le fasi del programmazione



METODI DI DESCRIZIONE DEGLI ALGORITMI

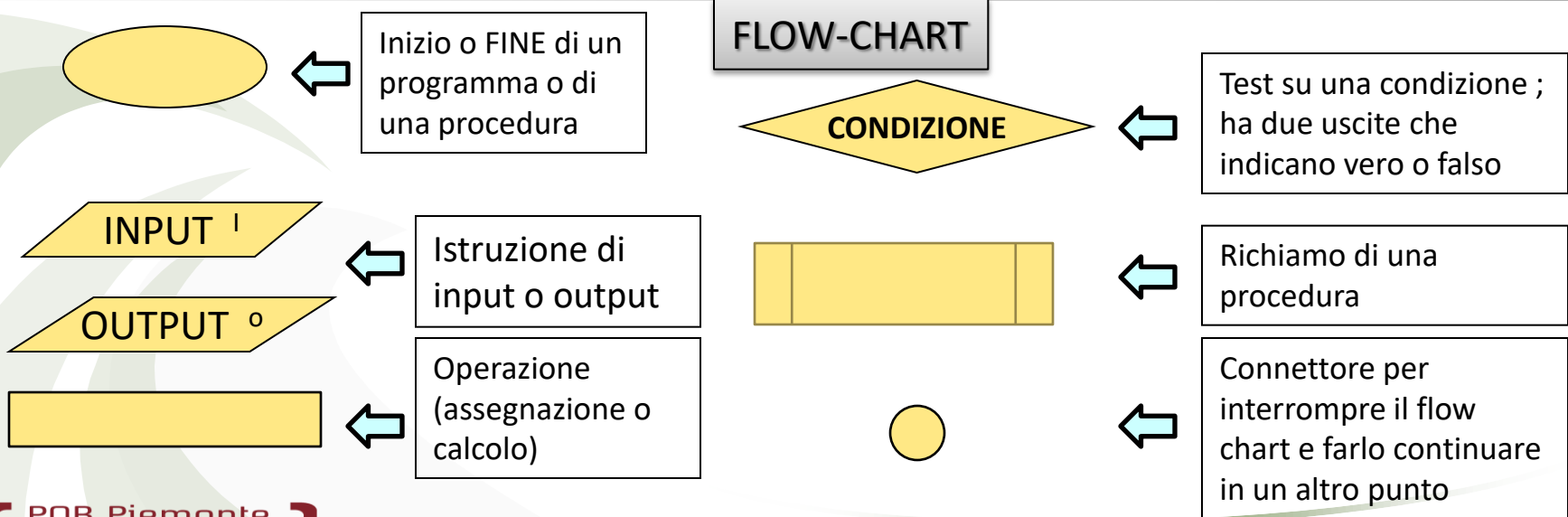
PSEUDOCODIFICA

Istruzioni descritte in linguaggio naturale

FLOW-CHART

Rappresentazione grafica con Diagrammi a blocchi o diagrammi di flusso

Sono due rappresentazioni diverse ma equivalenti



PSEUDOCODIFICA

INIZIO .. FINE



Inizio o fine di un programma
o di una procedura

Variabile <- espressione



Assegnazione di un valore a
una variabile

LEGGI variabile



Istruzione di input

SCRIVI testo o variabile



Istruzione di output

SE condizione **ALLORA**

istruzioni 1

ALTRIMENTI

istruzioni2

FINESE



Struttura di selezione

MENTRE condizione
istruzioni



Struttura ciclica con
valutazione in testa

FINEMENTRE

RIPETI

istruzioni

FINCHE condizione



Struttura ciclica con
valutazione in coda

PROGRAMMAZIONE STRUTTURATA

La programmazione strutturata permette di rendere più semplice la scrittura e la manutenzione di un programma.

La programmazione strutturata impedisce di usare le istruzioni di salto (GO TO) e permette al programmatore di avere in mente una struttura più chiara dell'algoritmo.

La programmazione strutturata richiede che un algoritmo sia descritto in modo comprensibile e per questo si devono usare le strutture di controllo fondamentali che sono: **SEQUENZA, SELEZIONE, ITERAZIONE**

Teorema di BHOM-JACOPINI o della programmazione strutturata

*Qualsiasi algoritmo può essere descritto usando soltanto le tre strutture di controllo fondamentali: **SEQUENZA – SELEZIONE- ITERAZIONE***

Non esistono regole generali che permettono di trovare la soluzione in modo automatico

Si chiama **PROBLEM SOLVING** la ricerca di soluzione di un algoritmo

Per ricercare una soluzione è importante :

- **Conoscere l'argomento del problema**
- **Usare algoritmi conosciuti per problemi analoghi**
- **Dividere il problema in sottoproblemi**

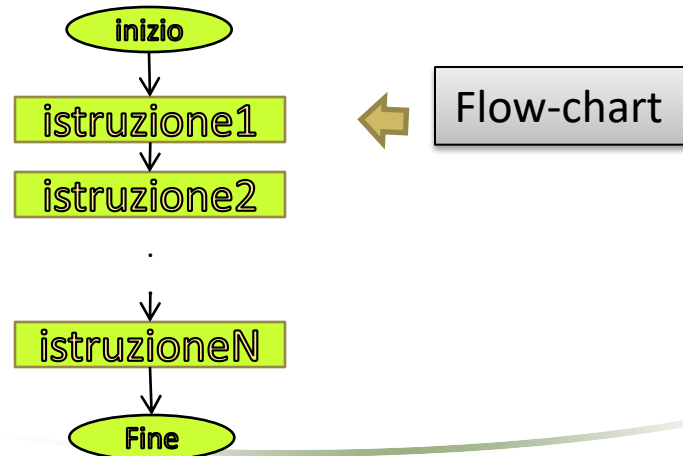
Un problema si dice COMPUTABILE se la sua soluzione può essere descritta mediante un algoritmo

STRUTTURA SEQUENZIALE o SEQUENZA

È la più semplice delle strutture.

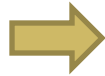
Con istruzione si intende una azione che deve essere eseguita. L'ordine con cui vengono eseguite deve essere rigoroso per trovare la soluzione. Le istruzioni sono quelle indicate nella pseudocodifica

Istruzione1
Istruzione 2
.
.
.
Istruzione N



STRUTTURE DI CONTROLLO

STRUTTURA DI SELEZIONE

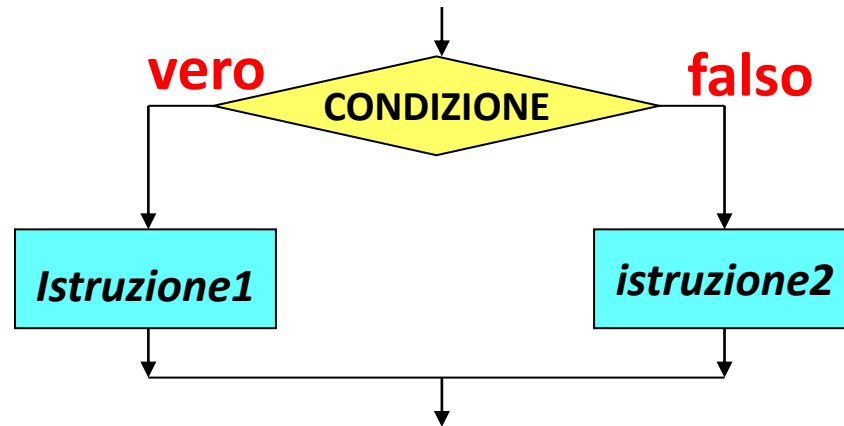


SELEZIONE BINARIA

È una struttura che valuta una condizione e a secondo del suo valore (vero o falso) viene svolta una istruzione o un'altra diversa.

SE condizione ALLORA
Istruzione 1
ALTRIMENTI
Istruzione 2
FINESE

La selezione può diventare
UNARIA se il ramo ALTRIMENTI
non presenta istruzioni



Flow-chart



STRUTTURE DI CONTROLLO

SELEZIONE MULTIPLA

NEL CASO CHE espressione SIA

caso 1:

istruzione1

caso 2:

istruzione 2

.....

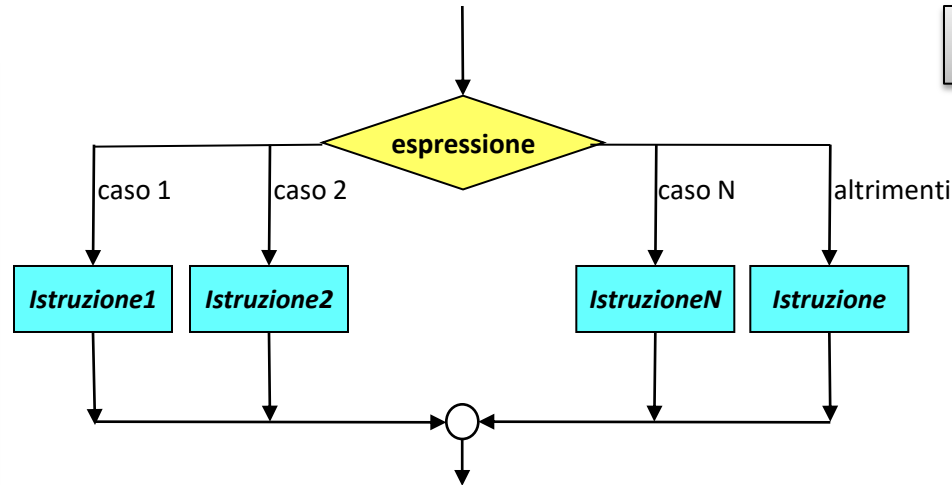
caso N:

istruzione N

ALTRIMENTI

Istruzione

FINECASO



Funzionamento:

Viene valutata l'espressione. Se il valore vale caso1 viene eseguita l'istruzione, o se il valore vale ccaso2 vie eseguita l'istruzione2, o se il valore casoN viene eseguita l'istruzione N. Nse non rientra in nuiesusn caso viene eseguita l'istruzione in Altrimenti. Solo un caso viene eseguito tra tutti quelli presenti

STRUTTURE DI CONTROLLO

STRUTTURA DI ITERAZIONE



RIPETIZIONE CON CONTROLLO IN TESTA

MENTRE condizione

Istruzione1

.....

Istruzione N

FINEMENTRE

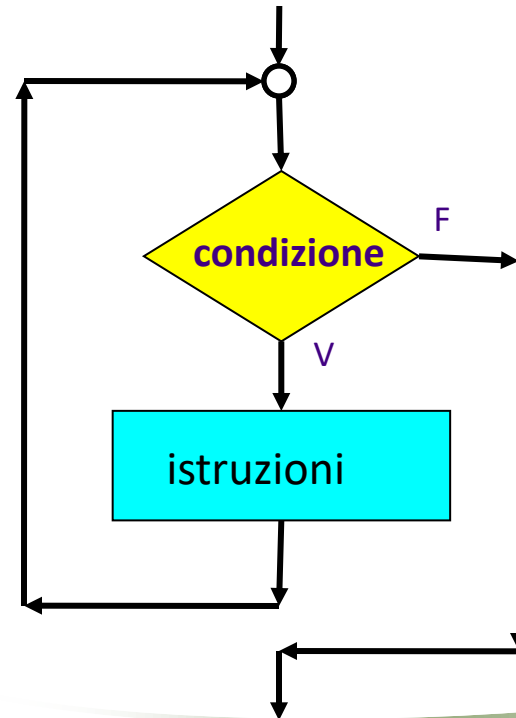
Funzionamento:

Viene valutata per primo la condizione.

Se la condizione è vera vengono eseguite le istruzioni.

Dopo avere eseguito tutte le istruzioni, viene di nuovo valutata la condizione, se è ancora vera si rieseguo le istruzioni. Le istruzioni si rieseguo sempre fino a quando la condizione è vera.

Il ciclo termina quando la condizione diventa falsa



Flow-chart

ESEMPIO

```
I=0  
MENTRE i<10  
    i = i + 2  
    SCRIVI i  
FINEMENTRE
```


STRUTTURE DI CONTROLLO

STRUTTURA DI ITERAZIONE



RIPETIZIONE CON CONTROLLO IN CODA

RIPETI

Istruzione1

.....

Istruzione N

FICHE condizione

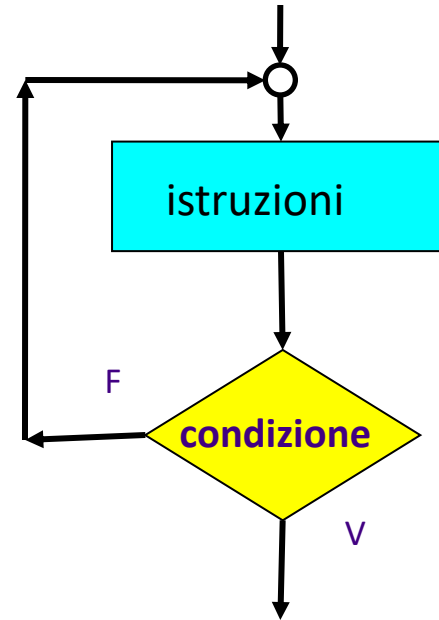
Funzionamento:

Vengono svolte per prima le istruzioni.

Poi si valuta la condizione. Se la condizione è falsa vengono rieseguite le istruzioni.

Poi viene di nuovo valutata la condizione, se è ancora falsa si rieseguo le istruzioni. Le istruzioni si rieseguo sempre fino a quando la condizione è falsa.

Il ciclo termina quando la condizione diventa vera.



Flow-chart



STRUTTURE DI CONTROLLO

STRUTTURA DI ITERAZIONE

PER indice Da 1 a N, passo k

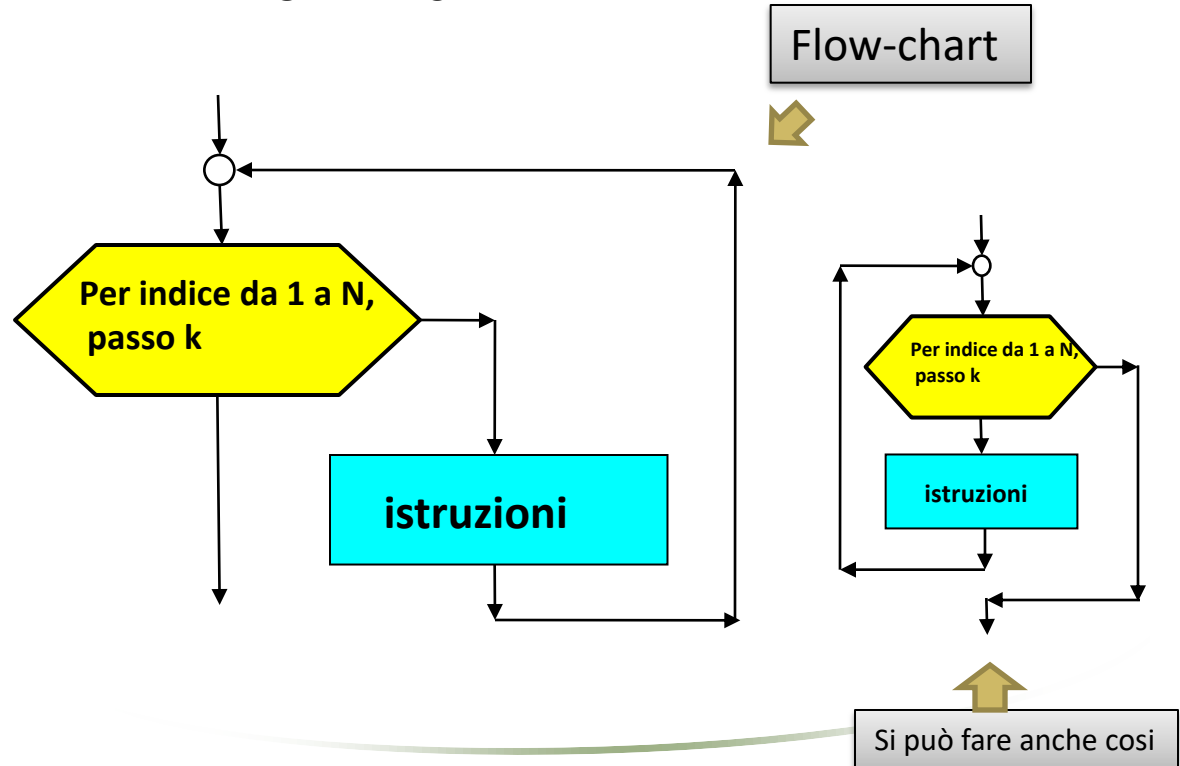
Istruzioni

FINEPER

Funzionamento:

l'indice viene posto uguale al primo valore, poi vengono svolte le istruzioni. Poi l'indice viene incrementato di k volte, se il valore non supera quello massimo N, vengono rieseguite le istruzioni e questo viene ripetuto finché l'indice non supera il valore massimo.

ITERAZIONE ENUMERATIVA



ESEMPIO DI STRUTTURA SEQUENZIALE

Esempio: Calcolare l'area e il perimetro di un quadrato di lato L

INIZIO

SCRIVI "Inserisci il valore del lato"

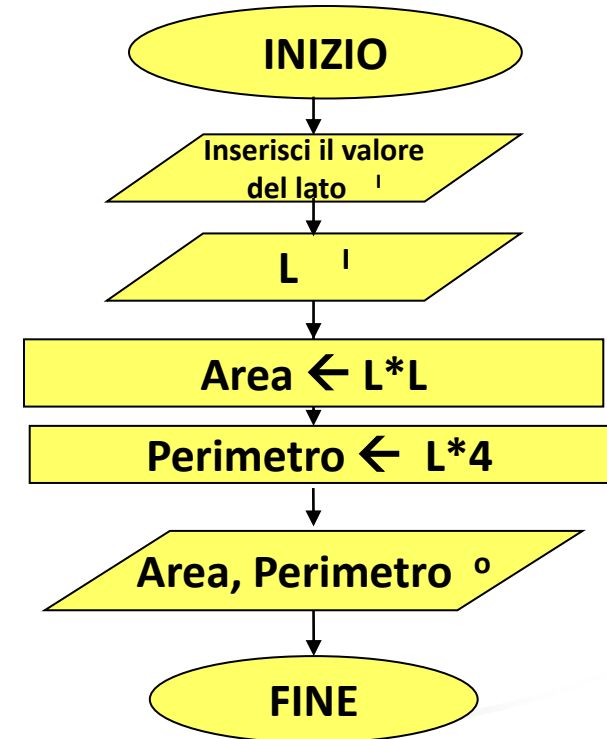
LEGGI L

Area = $L * L$

Perimetro = $L * 4$

SCRIVI Area, Perimetro

FINE



INIZIO

SCRIVI "inserisci nome della persona 1"

LEGGI P1

SCRIVI "inserisci nome della persona 2"

LEGGI P2

SE $P1 < P2$ ALLORA

SCRIVI " La stampa in ordine alfabetico dei nomi è"

SCRIVI P1;P2

ALTRIMENTI

SCRIVI " La stampa in ordine alfabetico dei nomi è"

SCRIVI P2;P1

FINESE

FINE

ESEMPIO DI STRUTTURA ITERATIVA

Dati N numeri in input calcolare la somma. Chiedere all'utente se vuole terminare il calcolo inserendo "si" o "no"

INIZIO

SCRIVI "SOMMA DI N NUMERI IN INPUT"

Somma=0

RIPETI

SCRIVI "inserisci un numero"

LEGGI N

somma=somma+N

SCRIVI "vuoi terminare? inserisci <si>"

LEGGI C

FINCHÉ C="si"

SCRIVI "la somma è: "

SCRIVI somma

FINE

Iterazione con Controllo in coda

INIZIO

SCRIVI "SOMMA DI N NUMERI IN INPUT"

Somma=0

SCRIVI "inserisci un numero"

C="si"

MENTRE C="si"

SCRIVI "inserisci un numero"

LEGGI N

somma=somma+N

SCRIVI "vuoi continuare? inserisci <si>"

LEGGI C

FINEMENTRE

SCRIVI "la somma è: "

SCRIVI somma

FINE

Iterazione con Controllo in testa

Esercizi da svolgere

1. Leggere un numero dall'input e stamparlo
2. Calcolare area di un quadrato di lato L
3. Calcolare l'area e il perimetro di un quadrato di lato L
4. Scambiare i valori di due variabili
5. Scrivere un algoritmo che stampa in ordine alfabetico i nomi di 2 persone presi dall'input
6. Dati due nomi di persona e le loro date di nascita ordinarli per nome o per data di nascita su richiesta dell'utente
7. Dati N numeri in input calcolare la somma. Chiedere all'utente se vuole terminare il calcolo inserendo "si" o "no"
8. Dati N numeri in input calcolare la somma e la Media. Chiedere all'utente se vuole terminare il calcolo inserendo "si" o "no"
9. Media di tutti i numeri messi in input. Richiedere se si vuole aggiungere altri numeri
10. Scrivere un algoritmo per calcolare se un numero è pari o dispari
11. Dati N numeri in Input, calcolare quanti sono pari e quanti sono dispari
12. Scrivere un programma accetti un numero in input e che controlli se l'input immesso è un numero
13. Dati n numeri in input, in lire, convertire in euro, e stampare il totale
14. Calcola il doppio dei primi 30 numeri naturali
15. Dati 15 Prezzi in input, calcolare lo sconto del 15% del il prezzo è maggiore di 125. Calcolare anche lo sconto totale e il prezzo totale scontato
16. Calcolare le radici di una equazione di secondo grado
17. Progettare un programma che dato in input le lettere dell'alfabeto visualizzare se è una vocale o una consonante. L'input termina su richiesta all'utente se vuole continuare ad inserire dati.

Variabili booleane e operazioni logiche

Congiunzione: **AND**

a	b	a AND b
F	F	F
F	V	F
V	F	F
V	V	V

Disgiunzione: **OR**

a	b	a OR b
F	F	F
F	V	V
V	F	V
V	V	V

Negazione: **NOT**

a	NOT a
F	V
V	F

Congiunzione: **AND**

Disgiunzione: **OR**

Negazione: **NOT**

OR esclusivo: **XOR**

OR esclusivo: **XOR**

a	b	NOT (a AND b)
F	F	V
F	V	V
V	F	V
V	V	F

a	b	NOT (a OR b)
F	F	V
F	V	F
V	F	F
V	V	F

a	b	a XOR b
F	F	F
F	V	V
V	F	V
V	V	F

EQUIVALENZE

Dire

equivale a

$A \neq B$

$\text{NOT}(A=B)$

$A \geq B$

$(A > B) \text{ OR } (A = B)$

$A \leq B$

$(A < B) \text{ OR } (A = B)$

$A > B$

$(\text{NOT}(A=B) \text{ AND } (\text{NOT}(A < B)))$

$A < B$

$(\text{NOT}(A=B) \text{ AND } (\text{NOT}(A > B)))$

Licenza



Quest'opera è distribuita con Licenza [Creative Commons Attribuzione - Non commerciale - Non opere derivate 4.0 Internazionale](https://creativecommons.org/licenses/by-nc-nd/4.0/).