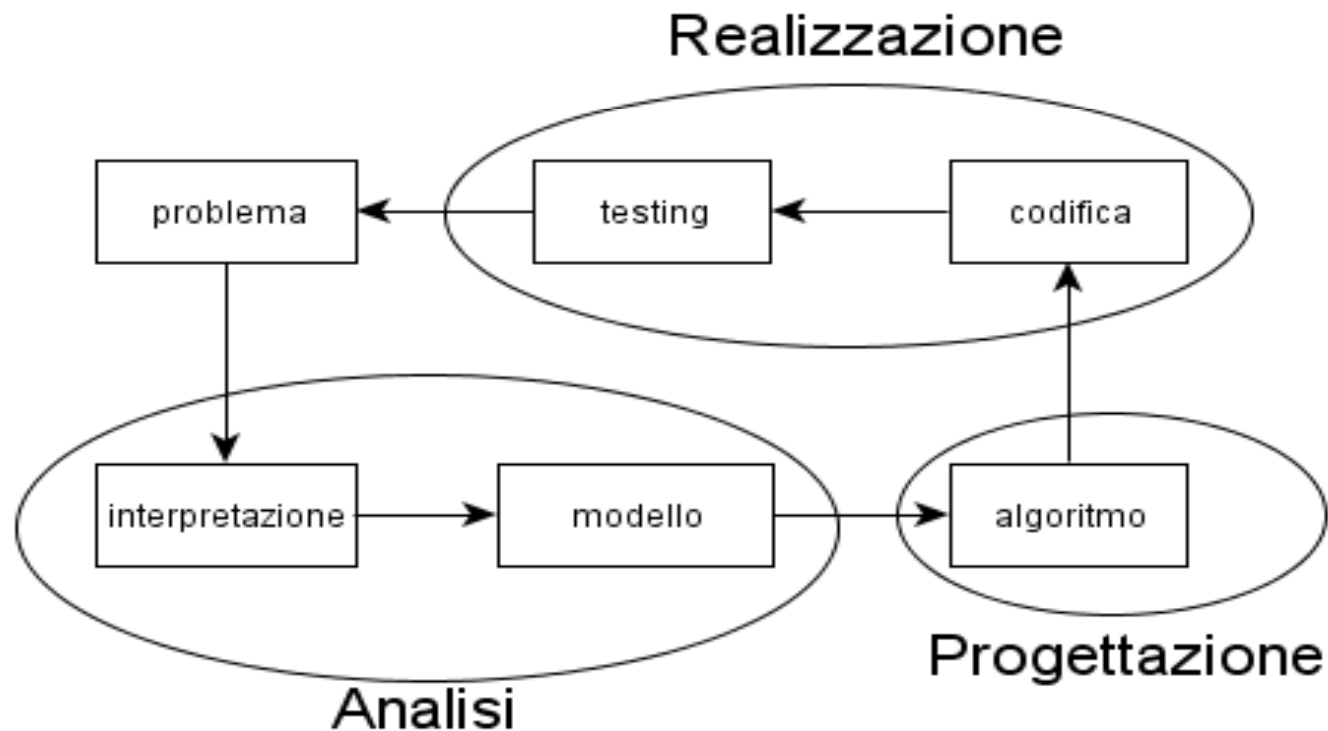




# ALGORITMI

# Cosa è un algoritmo

Un algoritmo è una sequenza di azioni che consente di pervenire alla soluzione di un problema mediante una sequenza finita di operazioni, completamente e univocamente determinate.



# Funzionalità di un Algoritmo

Possiamo immaginare l'algoritmo come un procedimento risolutivo che riceve dei dati in ingresso (input) esegue una qualche elaborazione e restituisce il risultato della trasformazione (output).



Uno stesso problema può essere risolto in modi diversi. Quindi ci possono essere più algoritmi che risolvono lo stesso problema.

# Proprietà di un algoritmo

## L' algoritmo deve essere:

- ◆ **Finito**, costituito cioè da un numero limitato di passi (le istruzioni sono in numero finito e vengono eseguite un numero finito di volte);
- ◆ **Definito (non-ambiguo)**, ogni istruzione deve essere elementare e deve consentire un'interpretazione univoca;
- ◆ **Eseguibile**, cioè la sua esecuzione deve essere possibile con gli strumenti di cui si dispone;
- ◆ **Deterministico**, ad ogni passo deve essere definita una ed una sola operazione successiva.
- ◆ **Generale**: un algoritmo risolve una classe di problemi: la descrizione dell'algoritmo non cambia quando cambiano i dati. Un algoritmo è quindi indipendente dai dati in ingresso.

# Rappresentazioni di un Algoritmo

## ◆ I Diagrammi a Blocchi (flow-chart)

- ha il pregio di evidenziare visivamente il flusso di esecuzione dell'algoritmo. Ha un impatto visivo molto forte.

## ◆ Pseudocodifica

- E' vicino al linguaggio naturale, utilizza un insieme di parole-chiave (parole che descrivono il linguaggio) che sono un sottoinsieme del nostro vocabolario.

# Formalismi

Un **formalismo** è insieme di formule e/o simboli usati in una teoria scientifica: Esempio: il formalismo della matematica.

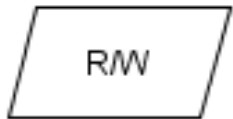
Così il flow-chart e la pseudocodifica sono dei formalismi usati per rappresentare algoritmi.

# Flow-chart

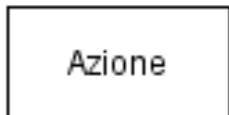
## ◆ Simboli convenzionali usati nel flow-chart:



- Simboli di inizio e fine algoritmo **begin/end**



- Simbolo di lettura **input** (Read) e stampa **output** (Write)



- Simbolo di **azione**



- Simbolo di scelta o **decisione**



CONNETTORE



SOTTOPROCESSO



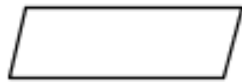
DOCUMENTO

Con questi simboli si può rappresentare un diagramma a blocchi. Ogni blocco del diagramma è la rappresentazione di un passo elementare.

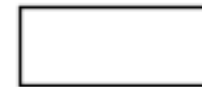
# Flow-chart

- Sono disponibili 4 operazioni base:

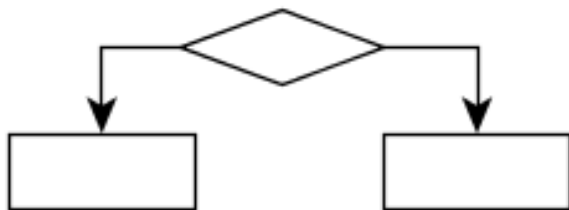
Input/output



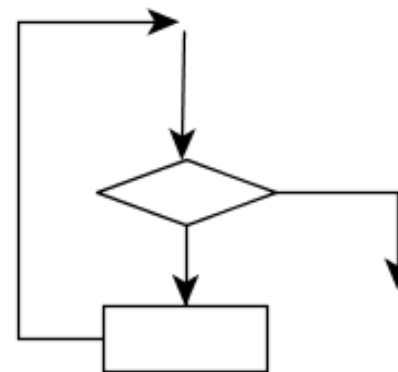
Esecuzione



Decisione (o selezione)



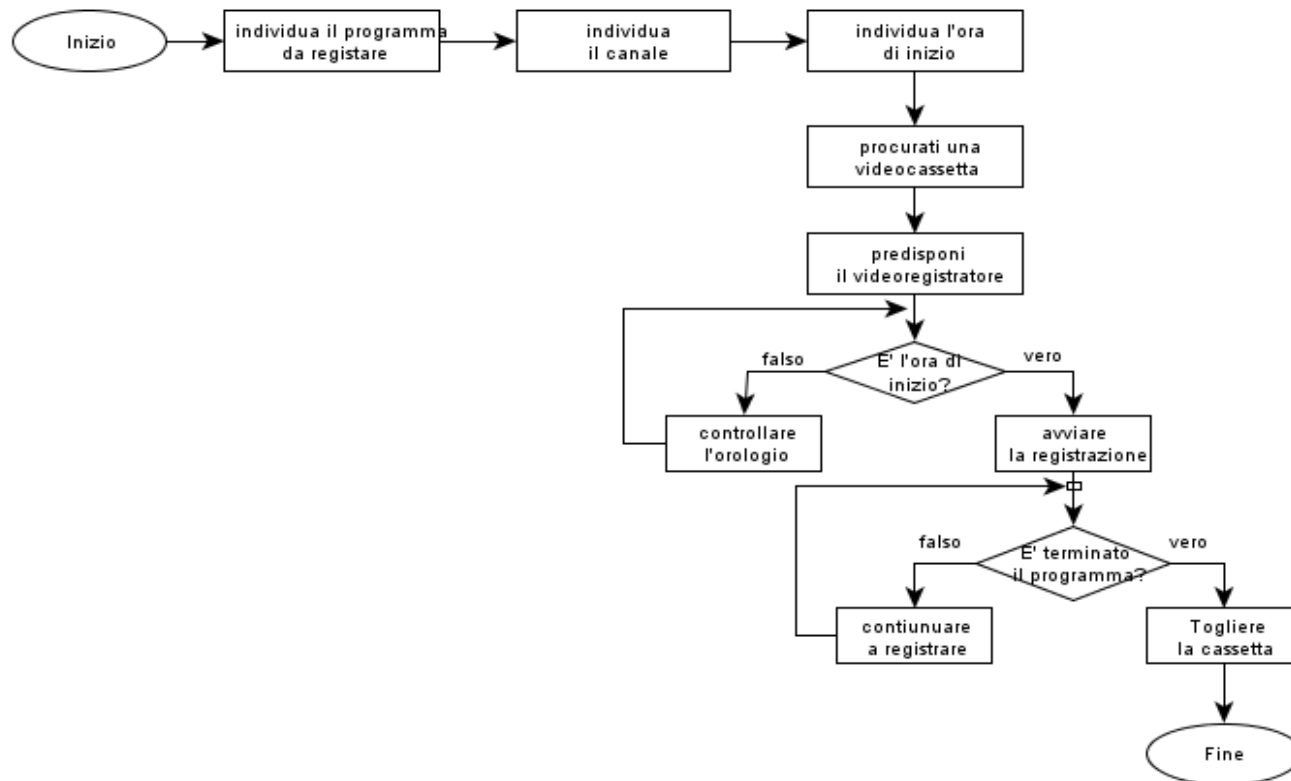
Iterazione (o ripetizione)





# Flow-chart (esempio)

- **Registrare un film con un videoregistratore.** Si vuole insegnare a un bambino a utilizzare un videoregistratore per registrare film.



# Flow-chart

## **Esercizio1. Olio nel motore.**

Per controllare il livello dell'olio nel motore è necessario aprire il cofano e, dopo aver individuato l'asta indicatrice, estrarla, pulirla e riporla; quindi estrarla nuovamente per effettuare la verifica del livello. Se tale livello è inferiore al minimo, si reinsertisce l'asta nell'alloggiamento, si svita il tappo e si aggiunge un po' d'olio, alternando successivi controlli finché si raggiunge una situazione soddisfacente. Quindi si ripone l'asta e si chiudono il tappo e il cofano dell'auto.

# Tecnica Top-down

Quando il problema è più articolato non è immediato trovare subito i passi elementari e la sua soluzione. Una tecnica che semplifica questo compito è la tecnica per raffinamenti successivi.

Il problema viene suddiviso in sottoproblemi.

Risolvendo i sottoproblemi si giunge alla soluzione del problema di partenza.

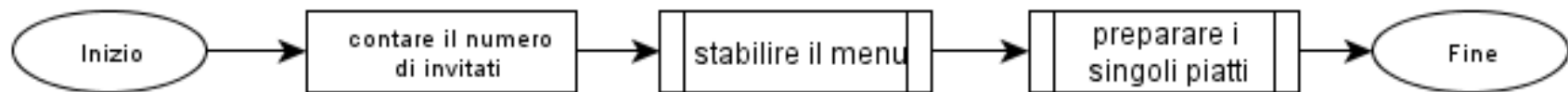
Tale procedimento di scomposizione è detta **tecnica top-down**. Cioè si parte dalla visione globale del problema (top=alto) e la si scompone in sottoproblemi autonomi che a loro volta sono composti da istruzioni elementari (down=basso).

# Tecnica Top-down (esempio)

## ESEMPIO. Pranzo di nozze.

Il problema può essere descritto come nel grafico. Il nuovo simbolo introdotto: 

Indica un sottoproblema. Cioè una azione non elementare che necessita di ulteriori passaggi elementari per essere dettagliata.



SOTTOPROCESSO

# Tecnica Top-down (esempio)

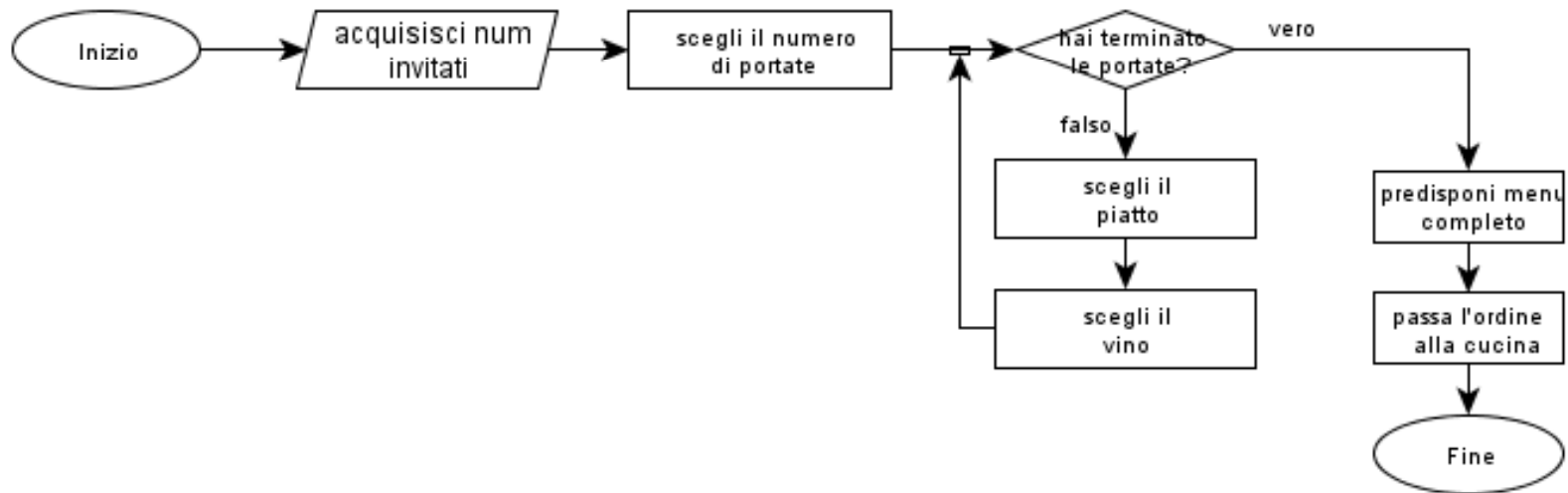
Ciascuno di questi problemi (che tra loro sono collegati) può essere risolto in modo indipendente.

Si passa al raffinamento successivo.

1. Il primo problema è un problema elementare, occorre acquisire il numero di invitati con un blocco di input.
2. Il secondo problema si articola nei seguenti passi:
  - scelta del numero di portate
  - per ogni portata, scelta della piatto,
  - scelta dei vini
3. Il terzo problema dovrebbe dare la descrizione di ogni ricetta. Ma se si suppone di passare l'ordine in cucina. L'esecuzione di una ricetta, per il cuoco è una competenza elementare.

# Tecnica Top-down (esempio)

Soluzione finale.



# Pseudocodifica

Il secondo formalismo per la descrizione di un algoritmo utilizza un sottoinsieme del linguaggio naturale. Utilizza le seguenti parole:

INIZIO/FINE

LEGGI (ACQUISISCI)/SCRIVI

SE ...ALLORA...ALTRIMENTI

MENTRE...ESEGUI

# Pseudocodifica

**Esempio. Registrare un film con il videoregistratore.**

**INIZIO**

**acquisisci** il programma da registrare

**acquisisci** il canale

**acquisisci** l'ora in cui inizia il programma

**acquisisci** una videocassetta

**acquisisci** il videoregistratore

**Mentre** "non è l'ora di inizio"

**Esegui**

    controlla l'orologio

**Mentre** il programma non è terminato

**Esegui**

    Continua a registrare

**Togli la cassetta**

**FINE**



# Flow-chart

## **Esercizio 1**

**Dato in input un numero, calcolarne il cubo se è maggiore di 0, il quadrato in caso contrario.**

## **Esercizio 2**

**Calcolare l'area di un rettangolo e quella di un quadrato**

## **Esercizio3. Buon giorno.**

Descrivi le azioni che vengono eseguite al mattino dal suono della sveglia fino all'uscita di casa.

## **Esercizio4. Secchi d'acqua.**

Sono presenti due secchi con capacità rispettivamente di 4 e 3 litri. Si riempia il primo secchio (quello da 4 litri) con soli 2 litri d'acqua. Descrivi con un algoritmo la soluzione.

# Flow-chart

## **Esercizio 5. Pizza Margherita.**

Si descriva la procedura per la preparazione di una pizza margherita.

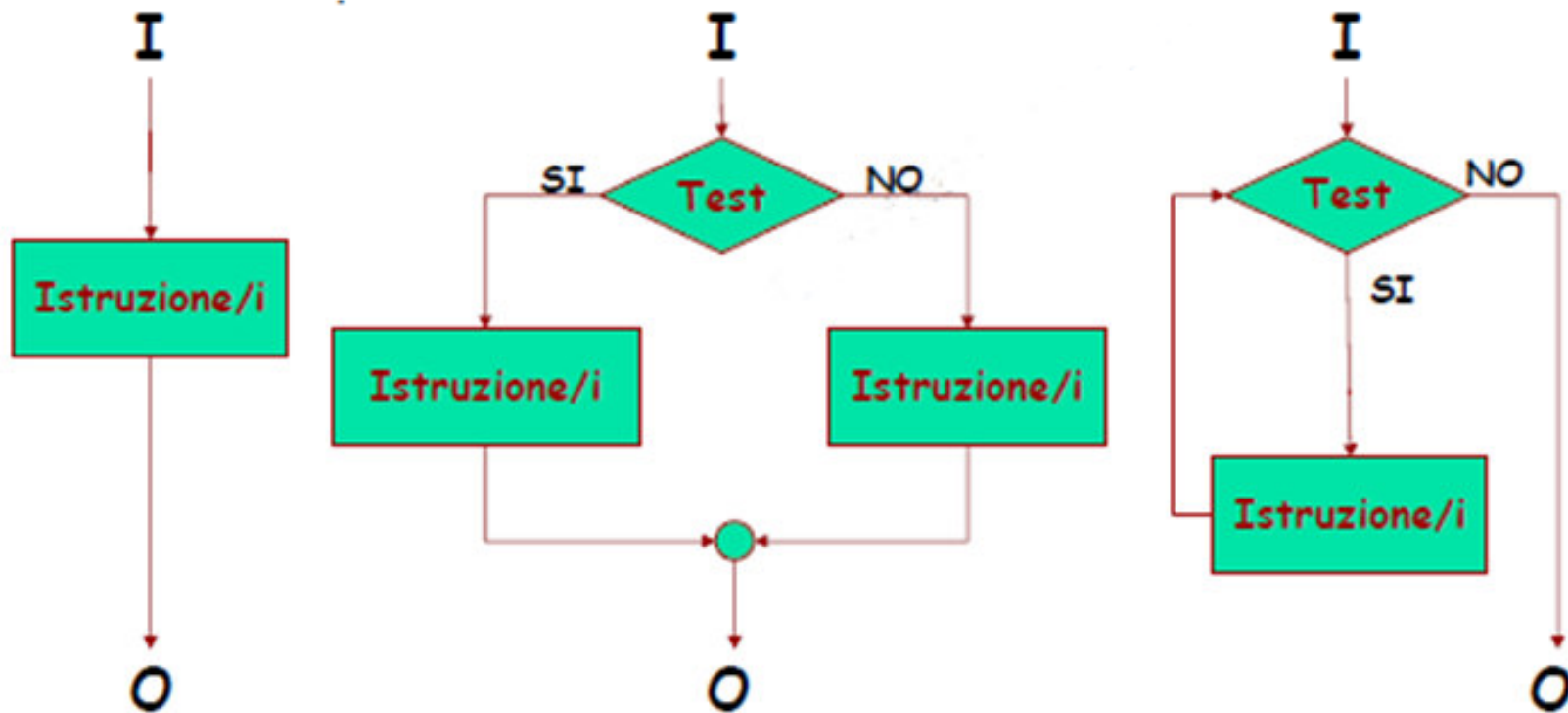
Dopo essersi procurati la ricetta della pizza ci si procura tutto il necessario, quindi si preparano la pasta e gli ingredienti. Si accende il forno (fuoco), si stende la pasta e la si farcisce con pomodoro e mozzarella. Quando il forno è alla temperatura giusta, si inforna la pizza e si attende la cottura. Quando la pizza è cotta, si estrae dal forno, si taglia e si mangia!

# SIMBOLI CONVENZIONALI (ESSENZIALI)

Sequenza

Decisione binaria (o istruzione If-Then-Else)

Ciclo, o ripetizione (o istruzione While-Do)



# TIPO DI CICLO

Un ciclo si può presentare in tre forme diverse, anche se tra loro equivalenti (cioè è possibile trasformare ogni ciclo negli altri due)

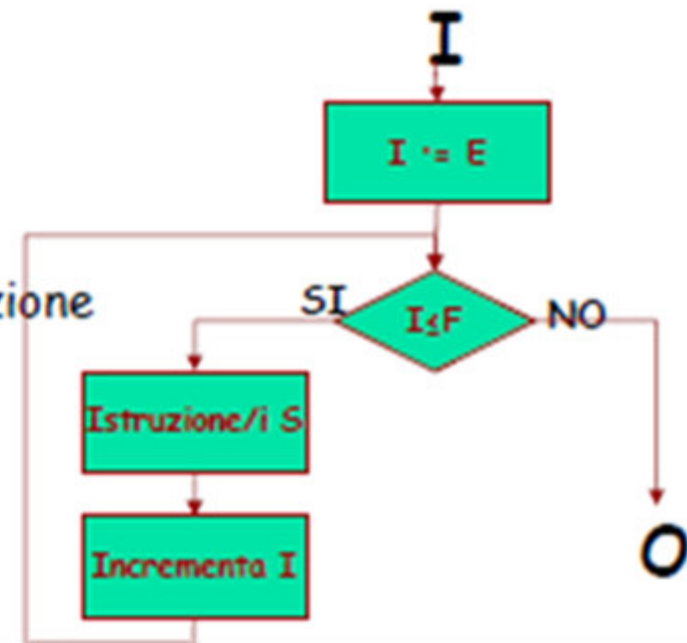
- ciclo a condizione iniziale (WHILE) - già incontrato
- ciclo a condizione finale (REPEAT) - non tratteremo
- ciclo a contatore (FOR)

1. esegue un insieme di istruzioni

2. se la variabile contatore  $I$ ,

- che parte da un valore iniziale  $E$
- e viene incrementata dopo ogni esecuzione

non supera il valore finale  $F$ .



# ESEMPIO MCD

