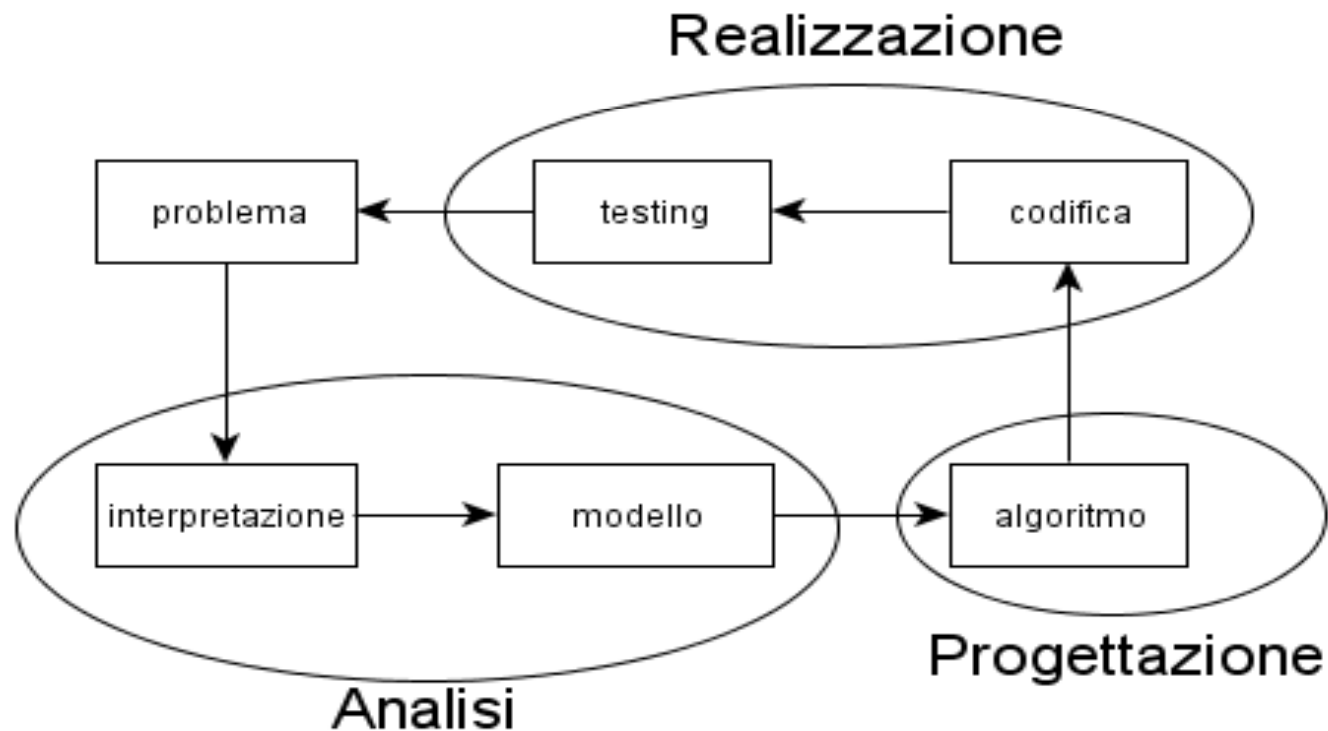




ALGORITMI

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

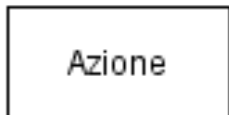
◆ Simboli convenzionali usati nel flow-chart:



- 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



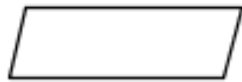
DOCUMENTO

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

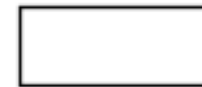
Flow-chart

- Sono disponibili 4 operazioni base:

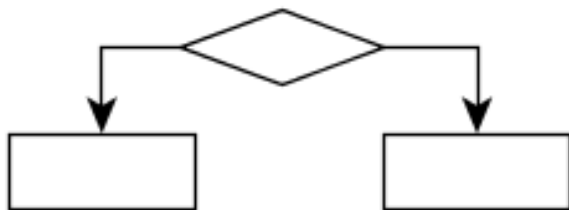
Input/output



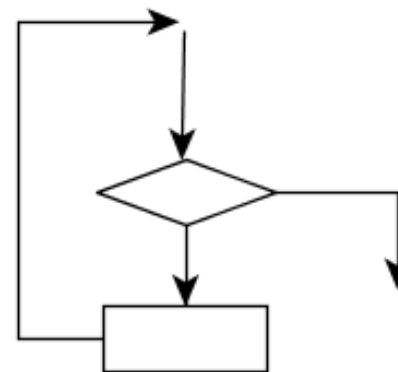
Esecuzione



Decisione (o selezione)

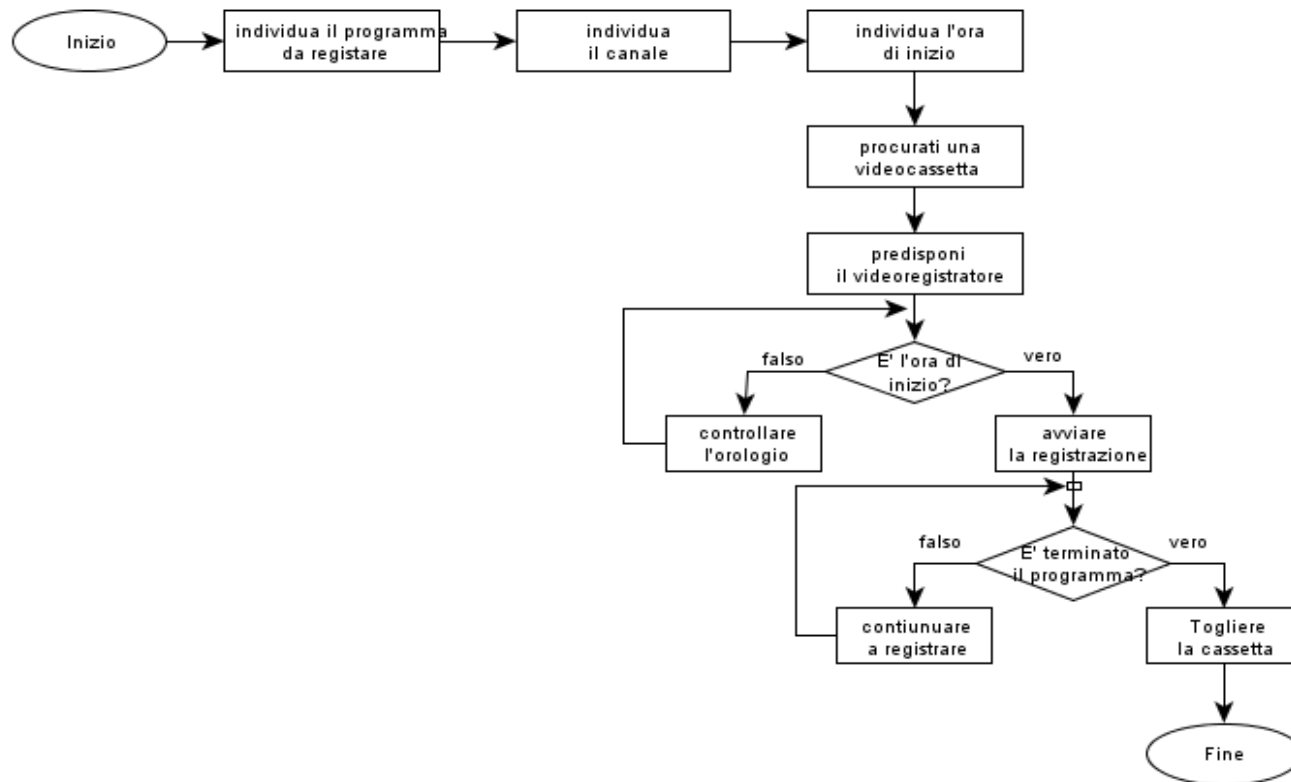


Iterazione (o ripetizione)



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.

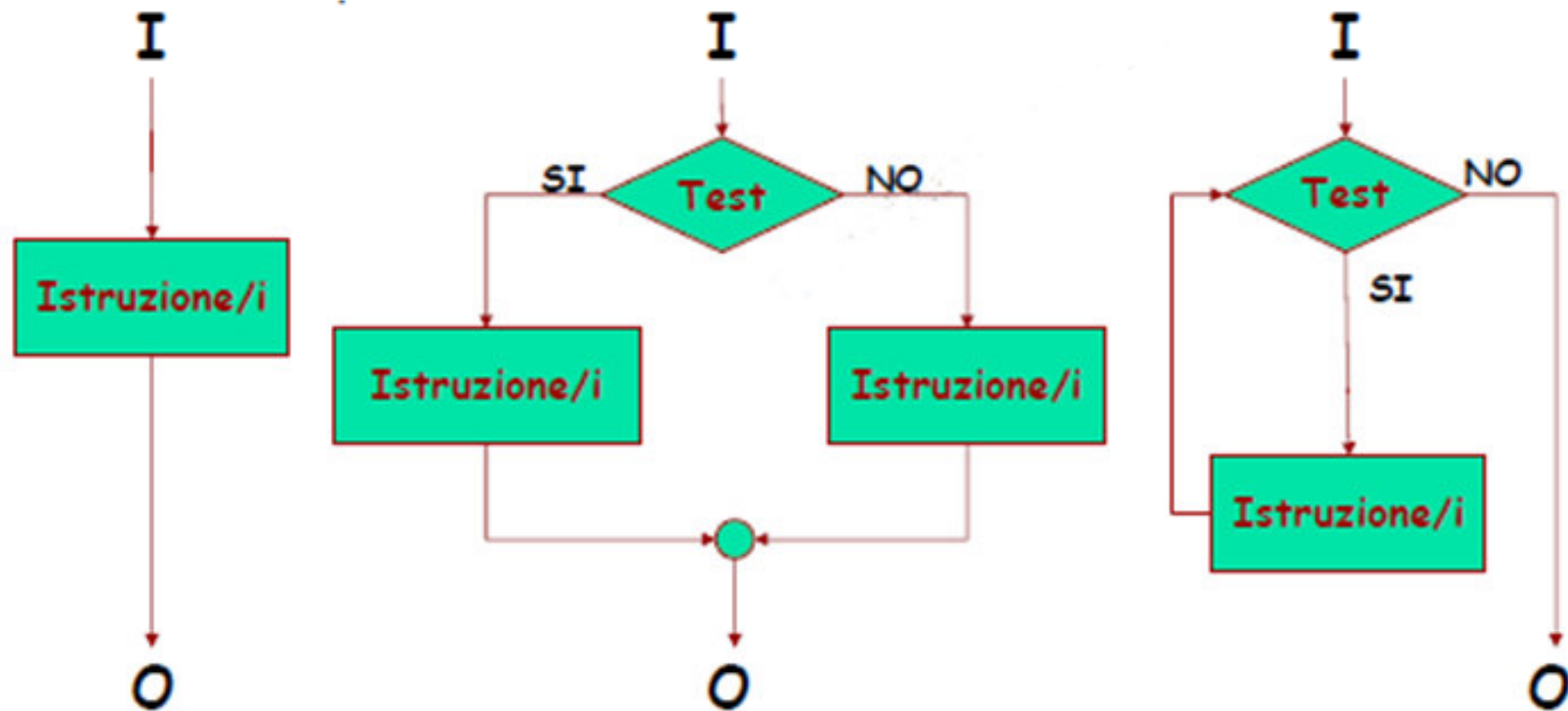
Per controllare il livello dell'olio nel motore è necessario aprire il cofano e, dopo aver individuato l'asta indicatrice, estrarla, pulirla e riporla; quindi estrarla nuovamente per effettuare la verifica del livello. Se tale livello è inferiore al minimo, si reinserisce l'asta nell'alloggiamento, si 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.

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)

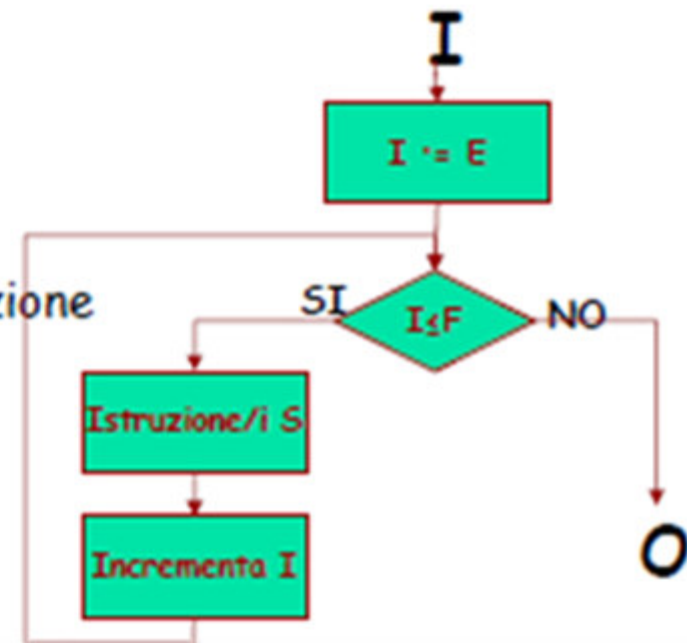
- ciclo a condizione iniziale (WHILE) - già incontrato
- ciclo a condizione finale (REPEAT) - non tratteremo
- ciclo a contatore (FOR)

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

