INFORMATICA

▪ APPROCCIO AI PROBLEMI

La realtà è ricca di situazioni problematiche. Ogni situazione problematica può essere tradotta in un problema che, analizzato in modo sistematico, ci consente di individuare i nodi di difficoltà in modo da arrivare alla soluzione.

Formulazione del problema

Prima di ricercare la soluzione di un problema occorre:

1. formulare in modo chiaro e non ambiguo il problema precisando: i reali obiettivi che si vogliono raggiungere, le regole, i dati espliciti e impliciti.
2. eliminare i dettagli inutili.

Modellizzazione del problema

La fase successiva è quella di rappresentare il problema con un modello nel quale vengono messi in evidenza solo gli aspetti utili e significativi per la risoluzione. In questa fase si ha un processo di **astrazione** che ci consente di costruire un procedimento risolutivo.
Qui compare la figura del **risolutore** che in base alle proprie esperienze e competenze costruisce un modello piuttosto che un altro. Il risolutore imposta la strategia per arrivare ai risultati; naturalmente deve tener conto delle capacità dell'esecutore, cioè di quali sono le istruzioni che quest'ultimo è in grado di comprendere e di eseguire.

Esecuzione

La figura dell'**esecutore** è in generale distinta da quella del risolutore e si occupa di eseguire una dopo l'altra le istruzioni contenute nel procedimento risolutivo.
Gli elaboratori elettronici sono degli esecutori in grado di effettuare in modo automatico una serie di istruzioni ricevute dall'uomo.
L'uomo in questo caso è il risolutore e la macchina è l'esecutore.

Verifica dei risultati

Questa fase consiste nel fare una verifica di attendibilità dei risultati ottenuti.

▪ DAL PROBLEMA ALL'ALGORITMO

Quando si vuole far eseguire un procedimento risolutivo, bisogna non solo conoscere le capacità dell'esecutore, ma anche saper dialogare con esso, dando istruzioni chiare e precise, utilizzando un linguaggio univoco e non ambiguo; si parla allora di **algoritmo.**

> Un **algoritmo** è *una sequenza finita di passi o azioni elementari per la risoluzione di una classe di problemi.*

Sequenza significa che è indicato l'ordine temporale; *elementare* significa che l'azione deve essere direttamente eseguibile dall'esecutore. *Classe di problemi*: problemi che differiscono solo per i dati iniziali.

Un algoritmo deve essere:

1. Finito sia come numero di istruzioni che come esecuzione, cioè l'elaborazione deve avere termine in un tempo finito
2. Illimitato: non deve essere posto un limite al numero di istruzioni di cui è composto
3. Non ambiguo: le istruzioni devono essere interpretate in modo univoco dall'esecutore
4. Deterministico: a partire dalle stesse condizioni iniziali deve produrre gli stessi risultati
5. Generale : deve risolvere tutti i problemi della stessa categoria
6. Deve fornire almeno un risultato.

- **DATI E ISTRUZIONI**

Le **istruzioni** di un algoritmo agiscono in generale su oggetti che sono i **dati.**

I **dati** possono essere: costanti se il loro valore non cambia nel tempo

variabili se durante l'esecuzione dell'algoritmo cambiano o possono cambiare il loro valore.
Ogni variabile è identificata da un **nome** e il valore della variabile rappresenta il contenuto della variabile
(nell'espressione 2 π r 2 e π sono delle costanti, r è una variabile)

Le **istruzioni** possono essere:

- **di assegnamento** : servono ad attribuire un valore alle variabili e sono del tipo:

A ← 5 ad A viene assegnato il valore 5
A ← A – 1 il valore di A viene decrementato di 1 e quindi diventa 4
B ← 3 - A + 6 / 5 a B viene assegnato il valore dell'espressione
3 - 4 + 6 / 5 cioè 0,2
dove A e B rappresentano nomi di variabili.

Quando ad una variabile viene assegnato un valore, il valore precedentemente contenuto nella variabile viene perso.
Una variabile si deve sempre **inizializzare** (assegnarle un valore iniziale), prima di usarla nella parte destra di un assegnamento. Ad es. l'assegnamento B ← 3 - A + 6 / 5 non ha alcun senso se alla variabile A non è stato attribuito nessun valore.

Ad ogni dato, cioè ad ogni variabile o costante deve essere associato un **tipo,** cioè l'insieme di valori che è possibile assegnare al dato in questione.

Se una variabile è numerica può essere di tipo **intero** se i valori che la variabile è destinata a contenere sono numeri interi
di tipo **reale** se i valori che la variabile è destinata a contenere sono numeri decimali finiti

Se una variabile è non numerica può essere:
di tipo **carattere** (variabili alfanumeriche) se i valori che la variabile è destinata a contenere sono singoli caratteri come ad es. le lettere dell'alfabeto, simboli di operazione, parentesi ect.
di tipo **stringa** (variabili alfanumeriche) se i valori che la variabile è destinata a contenere sono sequenze di caratteri e di parole come ad esempio il cognome e nome di una persona o il nome di una città etc.
di tipo **logico** (variabili booleane) se i valori che la variabile è destinata a contenere sono valori di verità (vero o falso).

- **di ingresso/uscita** (o input/output): servono ad acquisire dati di ingresso e a comunicare risultati e sono del tipo:

  leggi (A)
  acquisisci (B,C)
  queste istruzioni consentono di assegnare alle variabili di nome A, B, e C dei valori che vengono dati ad esempio da tastiera.

  Scrivi (A)
  Comunica (B,C)
  queste istruzioni consentono di comunicare dei risultati ad esempio vengono scritti su video i valori contenuti nelle variabili A, B e C.

- **di controllo decisionale** : servono a variare il flusso di esecuzione a seconda del verificarsi o meno di una determinata condizione e sono del tipo:

  se X > Y allora esegui
  l'istruzione 1,
  l'istruzione 2,
  …
  FINE

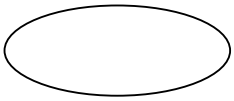
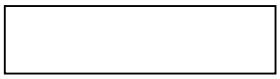
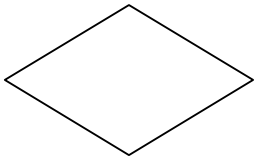
  altrimenti esegui
  l'istruzione 5,
  l'istruzione 6,
  …
  FINE

- **di controllo iterativo** : servono a ripetere un certo numero di istruzioni diverse volte, fino a quando non si verifica una certa condizione e sono del tipo:

  a) assegna ad A il valore 1
  b) aggiungi ad A il valore 2
  c) ***finché*** A non è maggiore di 100 torna all'istruzione **b)**
  d) FINE

DIAGRAMMI A BLOCCHI

I **diagrammi a blocchi** o **diagrammi di flusso** costituiscono un modo per rappresentare gli algoritmi.

Simbolo	Significato
	**INIZIO** / **FINE**
	**Istruzioni OPERATIVE** / **ASSEGNAMENTO**
	**Istruzioni di INPUT** / **OUTPUT**
	**Istruzioni di controllo DECISIONALE** / **ITERATIVO**
	**Flusso logico (**simbolo di connessione tra i blocchi**)**