

ALGORITMI - INDICE

- 1 - Cos'è un "algoritmo" 1
- 2 - Algoritmi e programmi 2, 3
- 3 - Diagrammi di flusso; programmazione strutturata 4, 5
- 4 - Esempi; lo pseudocodice (o "linguaggio di progetto") 6, 7, 8
- 5 - Esercizi 9 ... 15

ALGORITMI

1 - COS'È UN "ALGORITMO"

- a) Compra una confezione di filtri da camomilla
- b) riempi d'acqua una pentola
- c) metti la pentola sul fornello
- d) accendi il fornello
- e) attendi finché l'acqua bolle
- f) spegni il fornello
- g) metti un filtro in una tazza
- h) riempi la tazza con l'acqua della pentola
- i) schiaccia leggermente col cucchiaino il filtro e mescola finché l'acqua ha assunto un colore giallo uniforme
- j) toglì il filtro dalla tazza
- k) aggiungi un cucchiaino di zucchero e assaggia finché il liquido è dolce al punto giusto
- l) hai terminato



Quello che abbiamo scritto è un **algoritmo**:

una *sequenza di istruzioni*, finalizzata a risolvere il *problema* della *preparazione di una camomilla*.

- a) Leggi il valore di un numero naturale n
- b) se $n < 2$ allora
 - ┌ scrivi: "Il numero deve essere maggiore o uguale a 2: ridammelo";
 - └ leggi il valore di n
- c) poni $k=1$
- d) ripeti:
 - ┌ se n è divisibile per k allora poni $\text{massimodivisoretrovato}=k$;
 - └ incrementa di un'unità il valore di k
 finché $k=n$
- e) se $\text{massimodivisoretrovato}=1$ allora scrivi "Il numero dato è primo"
altrimenti scrivi "Il numero dato non è primo"
- f) hai terminato

Quello che abbiamo scritto è un altro **algoritmo**:

una *sequenza di istruzioni*, finalizzata a risolvere il *problema* di *stabilire se un dato intero è primo*.

COS'È, DUNQUE, IN GENERALE, UN ALGORITMO?

**E' UNA SEQUENZA DI ISTRUZIONI, DI "PASSI",
IL CUI SCOPO È LA RISOLUZIONE DI UN PROBLEMA.**

Le **ISTRUZIONI** di un algoritmo devono essere

- **PRIVE DI AMBIGUITÀ** nel loro **CONTENUTO**:
ogni istruzione deve essere interpretabile dall'esecutore in modo univoco;
- **PRIVE DI AMBIGUITÀ** nel loro **ORDINE**:
per ciascuna istruzione, deve essere chiaro qual è l'istruzione da eseguire dopo;
- **CONCRETAMENTE ESEGUIBILI**;
- **COMPRESIBILI DA PARTE DELL'ESECUTORE**, sia questo un essere umano o una macchina:
se un'istruzione non possedesse questo requisito,
occorrerebbe spezzarla in un gruppo di istruzioni più semplici;
- **IN NUMERO FINITO**, ed **ESEGUIBILI IN UN TEMPO FINITO**.

E il **problema risolto** deve essere di carattere **GENERALE**:

ad esempio, non sarebbe un algoritmo una sequenza di istruzioni che fosse capace esclusivamente di stabilire se è o non è primo un numero particolare, che so ... il 41.