Le fasi della programmazione

Ad un primo livello di astrazione l'attività della programmazione può essere suddivisa in quattro (macro) fasi principali.

1. Definizione del problema (specifica)
2. Individuazione di un procedimento risolutivo (algoritmo)
3. Codifica dell'algoritmo in un linguaggio di programmazione (codifica)
4. Esecuzione e messa a punto (esecuzione)

Specifica

- La prima fase della programmazione consiste nel comprendere e definire (specificare) il problema che si vuole risolvere.
- La specifica del problema può essere fatta in maniera più o meno rigorosa, a seconda del formalismo descrittivo utilizzato.
- La specifica di un problema prevede la descrizione dello stato iniziale del problema (dati iniziali, input) e dello stato finale atteso (i risultati, output).
- La caratterizzazione degli stati iniziale e finale dipende dal particolare problema in esame e dagli oggetti di interesse.

Esempi di specifica informale

1. Dati due numeri, trovare il maggiore.
2. Dato un elenco telefonico e un nome, trovare il numero di telefono corrispondente.
3. Data la struttura di una rete stradale e le informazioni sui flussi dei veicoli, determinare il percorso più veloce da A a B.
4. Scrivere tutti i numeri pari (a parte 2) che non sono la somma di due numeri primi (Congettura di Goldbach [1742]).
5. Decidere per **ogni** programma e per ogni dato in ingresso, se il programma C termina quando viene eseguito su quel dato.

Esempi (contd.)

Caratteristiche comuni ai problemi

informazioni in ingresso $\Longrightarrow$ informazioni in uscita

stato iniziale $\Longrightarrow$ stato finale

Osservazioni sulla formulazione dei problemi:

- la descrizione non fornisce un metodo risolutivo (es. 3)
- la descrizione del problema è talvolta ambigua o imprecisa (es. 2, con Mario Rossi che compare più volte)
- per alcuni problemi non è noto un metodo risolutivo (es. 4)
- esistono problemi per i quali è stato dimostrato che non può esistere un metodo risolutivo (es. 5 - *indecidibili*)

Noi considereremo solo problemi per i quali
è noto che esiste un metodo risolutivo *decidibili*.

Algoritmi

- Una volta specificato il problema, si determina un procedimento risolutivo dello stesso (algoritmo), ovvero un insieme di azioni da intraprendere per ottenere i risultati attesi.
- Il concetto di algoritmo ha origini molto lontane: l'uomo ha utilizzato spesso algoritmi per risolvere problemi di varia natura. Le capacità intellettive necessarie per risolvere un problema sono "codificate" nell'algoritmo.
- Solo in era moderna, ci si è posti il problema di caratterizzare problemi e classi di problemi per i quali è possibile individuare una soluzione algoritmica e solo nel secolo scorso è stato dimostrato che esistono problemi per i quali non è possibile individuare una soluzione algoritmica.

Proprietà di un algoritmo

La descrizione di un procedimento risolutivo può considerarsi un algoritmo se rispetta alcuni requisiti essenziali, tra i quali:

Finitezza: un algoritmo deve essere composto da una sequenza finita di passi elementari.

Non-ambiguità: l'esecutore deve poter interpretare in modo univoco ogni singola azione.

Eseguibilità: il potenziale esecutore deve essere in grado di eseguire ogni singola azione in tempo finito con le risorse a disposizione.

Tipici procedimenti che non rispettano alcuni dei requisiti precedenti:

- le ricette di cucina: *aggiungere sale q.b.* - non rispetta 3)
- le istruzioni per la compilazione della dichiarazione dei redditi (!)

Proprietà di un algoritmo (cont.)

Sarà poi opportuno anche valutare l'efficienza di un algoritmo.

Se dato un elenco telefonico e un nome, per trovare il numero di telefono corrispondente guardo e confronto un nome dopo l'altro, ci metto molto più tempo che procedendo nel modo seguente:

- apro a metà e leggo il nome in cima;
- se trovo il nome che cerco ho finito, altrimenti
- se il nome cercato viene prima in ordine lessicografico, allora cerco nella prima metà;
- altrimenti cerco nella seconda metà;
- andando avanti nello stesso modo fino a trovare il nome o a decidere che non compare nell'elenco.

Quanto tempo ci metterò nei due casi?

Codifica

- Questa fase consiste nell'individuare una rappresentazione degli oggetti di interesse del problema ed una descrizione dell'algoritmo in un opportuno linguaggio noto all'esecutore.
- Nel caso in cui si intenda far uso di un elaboratore per l'esecuzione dell'algoritmo, quest'ultimo deve essere tradotto (codificato) in un opportuno linguaggio di programmazione. Il risultato in questo caso è un programma eseguibile per il calcolatore.
- Quanto più il linguaggio di descrizione dell'algoritmo è vicino al linguaggio di programmazione scelto, tanto più semplice è la fase di traduzione e codifica. Se addirittura il linguaggio di descrizione coincide con il linguaggio di programmazione, la fase di traduzione è superflua.

Codifica (cont.)

I linguaggi di programmazione forniscono strumenti linguistici per rappresentare gli algoritmi sotto forma di programmi che possano essere compresi da un calcolatore.

In particolare dobbiamo rappresentare nel linguaggio di programmazione

- l'algoritmo $\Longrightarrow$ programma
- le informazioni iniziali $\Longrightarrow$ dati in ingresso
- le informazioni utilizzate dall'algoritmo $\Longrightarrow$ dati ausiliari
- le informazioni finali $\Longrightarrow$ dati in uscita

In questo corso impareremo a codificare algoritmi utilizzando il linguaggio di programmazione denominato C.

Esecuzione

- La fase conclusiva consiste nell'esecuzione vera e propria del programma.
- Spesso questa fase porta alla luce errori che possono coinvolgere ciascuna delle fasi precedenti, innescando un procedimento di messa a punto tale da richiedere la revisione di una o più fasi (dalla specifica, alla definizione dell'algoritmo, alla codifica di quest'ultimo).
- Nel caso dei linguaggi di programmazione moderni, vengono forniti strumenti (denominati di solito debugger) che aiutano nella individuazione degli errori e nella messa a punto dei programmi.

Esempi

Negli esempi che seguono, utilizziamo un linguaggio pseudo-naturale per la descrizione degli algoritmi.

Tale linguaggio utilizza, tra l'altro, le comuni rappresentazioni simboliche dei numeri e delle operazioni aritmetiche.

Quanti hanno lo zaino in questa aula?

Specifica

Input: sequenza di postazioni e zaini in aula

Output: il numero di zaini

Algoritmo

Un semplice algoritmo è il seguente:

Passo 1.	Associa zero al *contatore* N
Passo 2.	Per ogni zaino che vedi **ripeti**
Passo 2.1.	Incrementa N di 1
Passo 3.	Ottieni il risultato dell'operazione in N

È corretto?

Quanti hanno lo zaino in questa aula? (cont.)

Specifica

Input: sequenza di postazioni e zaini in aula

Output: il numero di zaini

Algoritmo

Un algoritmo più veloce è il seguente:

Passo 1.	Associa zero al *contatore* N
Passo 2.	Per ogni coppia di zaini che vedi **ripeti**
Passo 2.1.	Incrementa N di 2
Passo 3.	Ottieni il risultato dell'operazione in N

È corretto?

Quanti hanno lo zaino in questa aula? (cont.)

Specifica

Input: sequenza di postazioni e zaini in aula

Output: il numero di zaini

Algoritmo

Un algoritmo più veloce ma corretto è il seguente:

Passo 1.	Associa zero al *contatore* N
Passo 2.	Per ogni coppia di zaini che vedi **ripeti**
Passo 2.1.	Incrementa N di 2
Passo 3.	Se rimane uno zaino
Passo 3.1.	Incrementa N di 1
Passo 4.	Ottieni il risultato dell'operazione in N

È corretto?

Problema 1: Calcolo del prodotto di due interi positivi

Specifica

Input: due valori interi positivi A e B

Output: il valore di A×B

Algoritmo

Se l'esecutore che scegliamo è in grado di effettuare tutte le operazioni di base sui numeri, un semplice algoritmo è il seguente:

Passo 1.	Acquisisci il primo valore, sia esso A
Passo 2.	Acquisisci il secondo valore, sia esso B
Passo 3.	Ottieni il risultato dell'operazione A×B

Problema 1 (cont.)

Codifica

Un esecutore che rispetta le ipotesi precedenti è una persona dotata di una calcolatrice tascabile. La codifica dell'algoritmo in questo caso può essere allora la seguente:

Passo 1.	Digita in sequenza le cifre decimali del primo valore
Passo 2.	Digita il tasto $*$
Passo 3.	Digita in sequenza le cifre decimali del secondo valore
Passo 4.	Digita il tasto $=$

Esecuzione

Problema 1 (cont.)

- Supponiamo ora che l'esecutore scelto sia in grado di effettuare solo le operazioni elementari di somma, sottrazione, confronto tra numeri.
- È necessario individuare un nuovo procedimento risolutivo che tenga conto delle limitate capacità dell'esecutore

Algoritmo 2

Passo 1.	Acquisisci il primo valore, sia esso A
Passo 2.	Acquisisci il secondo valore, sia esso B
Passo 3.	Associa 0 ad un terzo valore, sia esso C
Passo 4.	**Finché** B>0 **ripeti**
Passo 4.1.	Somma a C il valore A
Passo 4.2.	Sottrai a B il valore 1
Passo 5.	Il risultato è il valore C

Problema 1: (cont.)

Un esecutore che rispetta le ipotesi precedenti è un bimbo in grado di effettuare le operazioni richieste (somma, sottrazione e confronto) e di riportare i risultati di semplici calcoli su un quaderno.

Codifica

Passo 1.	Scrivi il primo numero nel riquadro A
Passo 2.	Scrivi il secondo numero nel riquadro B
Passo 3.	Scrivi il valore 0 nel riquadro C
Passo 4.	Ripeti i seguenti passi finché il valore nel riquadro B è maggiore di 0: - calcola la somma tra il valore in A e il valore in C - scrivi il risultato ottenuto in C - calcola la differenza tra il valore in B ed il numero 1 - scrivi il risultato ottenuto in B
Passo 5.	Il risultato è quanto contenuto nel riquadro C.