

Introduzione alla programmazione

Università degli Studi di Brescia

Docente: Massimiliano Giacomini

VEDIAMO UN ESEMPIO FACILE... FORSE...

- PROVATE A SVOLGERLO INDIVIDUALMENTE
- RIGUARDO AL LINGUAGGIO C, BASTA CONOSCERE:
 - la definizione e l'uso delle variabili intere
 - if-else
 - strutture iterative (basta anche solo while)
- FORSE VI SEMBRA FACILE (EFFETTIVAMENTE LO E')
PERO' PROVATE AD ARRIVARE IN FONDO...
- PROVATE AD OTTENERE UN CODICE "ORDINATO"...

Dal tema d'esame del 13 gennaio 2009 [Ing. dell'Informazione]

Si sviluppi un programma in linguaggio C che, ricevendo in ingresso una sequenza di lunghezza arbitraria di almeno due numeri interi diversi da zero, terminata da uno zero, produca in uscita i due valori minimi letti in ingresso (escluso l'ultimo zero).

Ad esempio, ricevendo in ingresso la sequenza

7 2 19 4 45 3 7 9 3 0

produce in uscita 2 3

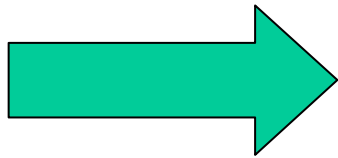
Altro esempio: ricevendo in ingresso la sequenza

7 2 19 4 2 3 7 9 3 0

produce in uscita 2 2

[10]

- FORSE NON E' STATO PROPRIO COSI' FACILE...
- FORSE IL CODICE NON E' COSI' "ORDINATO"...

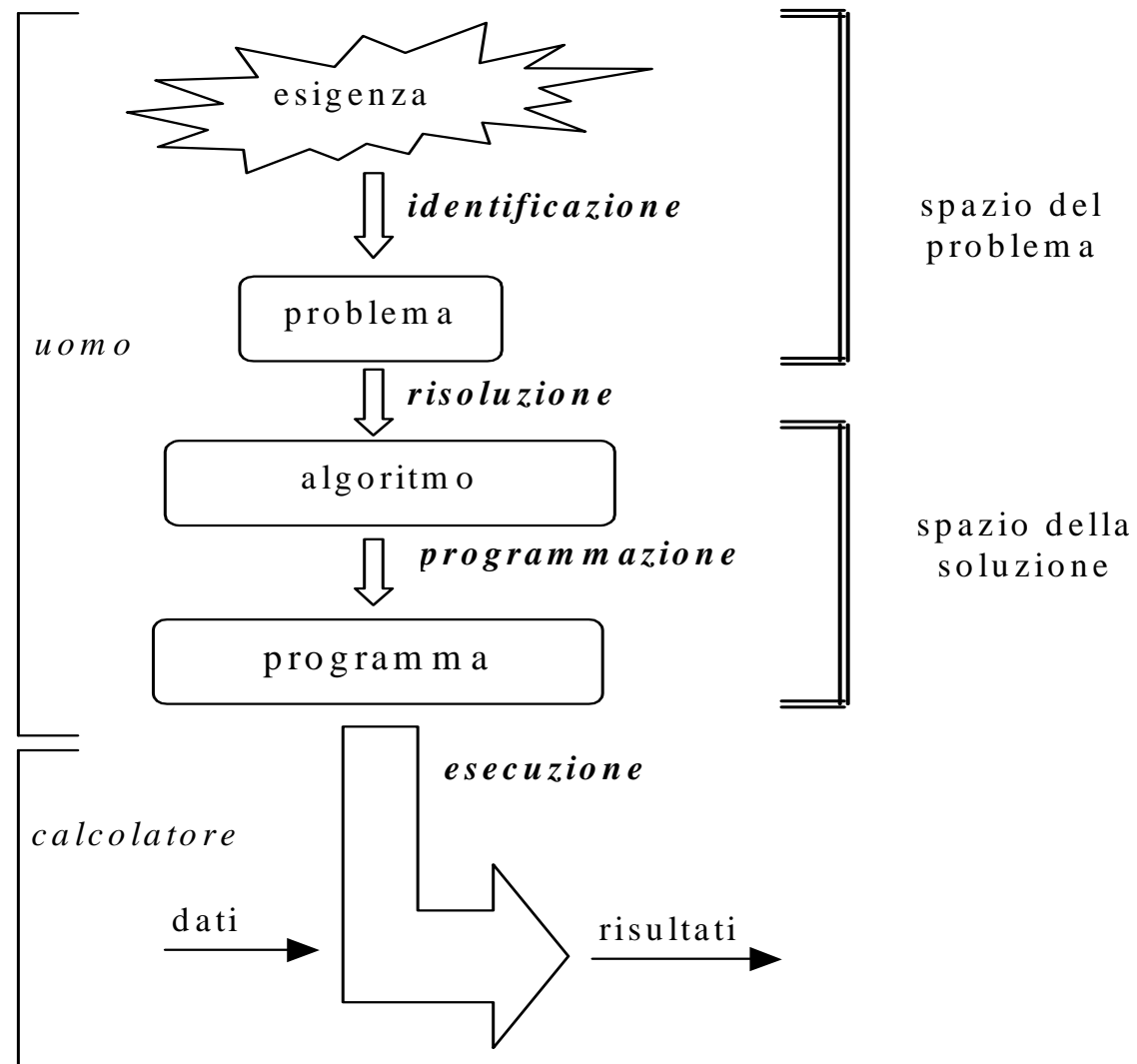


SAPER PROGRAMMARE

≠

CONOSCERE IL C

LA RISOLUZIONE DI PROBLEMI CON IL CALCOLATORE



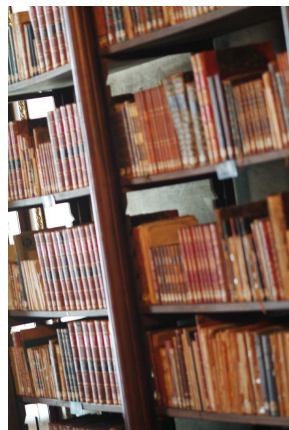
Identificazione

Comprensione del problema: eliminando ogni ambiguità nella sua formulazione, focalizzando gli obiettivi, evidenziando i dati impliciti ed espliciti del problema

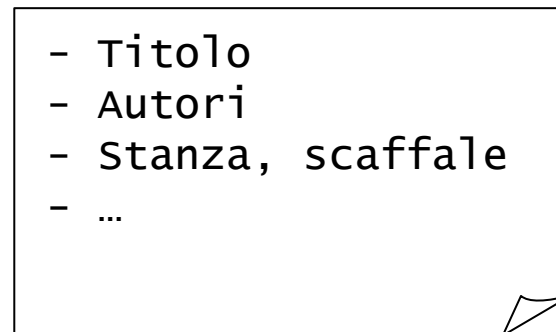
Modellazione del problema: creando un modello

↓
Rappresentazione semplificata della realtà, che rappresenta un' astrazione del problema

Esempio: l'archivio di una biblioteca



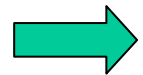
Mondo reale



Modello

Risoluzione e programmazione

- Dal problema, ben formulato e “modellato” a una sequenza di “*istruzioni elementari*” (eseguibili dal calcolatore), p.es. istruzioni C



- Risolvere un problema:

SCOMPOSIZIONE IN SOTTO-PROBLEMI

- I sotto-problemi potrebbero dover essere a loro volta scomposti in *sotto-sotto-problemi* e così via
- La scomposizione deve giungere fino ai **problemi elementari (o primitivi)** che corrispondono alle istruzioni elementari
(espresse nel linguaggio di programmazione usato)

UN ESEMPIO INFORMALE

- Gestione di una biblioteca in cui:
 - I libri sono disposti sugli scaffali
 - La posizione di ogni libro è data da: numero dello scaffale e posizione nello scaffale
 - Esiste uno schedario ordinato in cui ogni scheda riporta le seguenti informazioni:
 - **Cognome e nome dell'autore**
 - **Titolo del libro**
 - **Data di pubblicazione**
 - **Numero dello scaffale in cui si trova**
 - **Posizione nello scaffale**

Problema: richiesta di un libro da parte di un cliente

- Come procedo?
 1. Ricevo la richiesta dall'utente
 2. Prelevo il libro
- Il secondo passo va dettagliato, ovvero va scomposto in sotto-problemi (*procedura per raffinamenti successivi o top-down*)

Il problema diventa:

1. Ricevo la richiesta dell'utente
2. Cerco la scheda del libro nello schedario
3. Mi segno numero dello scaffale e posizione nello scaffale
4. Cerco lo scaffale
5. Cerco il libro nella sua posizione all'interno dello scaffale
6. Prelevo il libro

Non è un problema elementare!

Cercare la scheda...

- Scompongo in sotto-sotto-problemi:
 1. Prendo la prima scheda
 2. Il titolo, l'autore e la data corrispondono a quelli del libro che sto cercando? *Se* sì *allora* ho individuato la scheda, altrimenti passo alla scheda successiva e ripeto il controllo
 3. Se le schede sono esaurite, allora il libro non esiste
- Esistono metodi più efficienti per risolvere lo stesso problema... come fareste voi?

Un metodo più efficiente...

1. Esamino la scheda centrale dello schedario
2. Se la scheda corrisponde al libro cercato allora termino la ricerca
3. Altrimenti cerco (con lo stesso metodo) nella metà inferiore o superiore dello schedario a seconda che il libro cercato preceda o segua il libro indicato sulla scheda

In realtà il passo 2 deve “accorgersi” anche se il libro non esiste, diventa:

“se la scheda corrisponde al libro cercato oppure se la parte di schedario da esaminare è vuota allora termino la ricerca”

RISOLVERE I TEMI D'ESAME

UN PARAGONE...

SVILUPPO DI UN TEMA



- **Leggere bene la traccia
(evitare di andare “fuori tema”)**
- **Pensare a cosa si sa e a cosa si può scrivere,
quali tematiche toccare, ecc.**
- **Pensare (meglio ancora scrivere!) uno schema:
come organizzare le informazioni?**
- **SOLO DOPO, stendere il tema
(è possibile modificare lo schema)**

ERRORE TIPICO: buttarsi a scrivere tutto “di getto”

SVILUPPO DI UN PROGRAMMA C

- **Capire il problema**
(eventualmente: provare a risolvere qualche istanza!)
- **Pensare a come risolverlo (come faremmo noi?)**
- **Arrivare ad un algoritmo per passi successivi, partendo da una descrizione più astratta e poi scomponendola in passi sempre più elementari...**
[NB: è possibile tornare indietro e modificare lo schema di partenza]
- **... fino ad arrivare al linguaggio C**

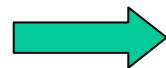
ERRORE TIPICO: buttarsi a scrivere il codice “di getto”

LE COSE PIU' DIFFICILI DA IMPARARE:

- Applicare il metodo dei raffinamenti successivi “dove veramente serve”: tra *scanfe* e “*acquisisci n*” non c'è molta differenza, l'astrazione deve semplificare la prima formulazione del problema permettendoci di concentrarci sul cuore del problema stesso
- Affrontare prima i casi più generali e poi le eccezioni
(ERRORE TIPICO: cominciare il programma con una sequela di if...)
- Imparare a “tornare indietro” fin dalle prime fasi del processo di scomposizione per raffinamenti successivi, per giungere ad una stesura più semplice o più ordinata del codice
- Dare un significato preciso alle variabili e attenersi nella stesura del codice a questo significato (non procedere per tentativi!)

COME IMPARARLE

- Purtroppo o per fortuna, non esistono regolette da applicare
- Come nello sport:
 - bisogna fare *esercizio*
 - bisogna abituarsi ad applicare gli accorgimenti già quando si affrontano esercizi semplici: solo così la *tecnica* verrà poi applicata “spontaneamente” negli esercizi più complicati



Bruciare le tappe non porta lontano,
viceversa applicare la tecnica fin da subito...