

Una lezione (quasi) universitaria di Informatica.
Liceo Scientifico Statale "Galileo Ferraris".

Torino, 27 aprile 2007

Parte I

Elio Giovannetti
Dipartimento di Informatica
Università di Torino



Quest' opera è pubblicata sotto una Licenza Creative Commons
Attribution-NonCommercial-ShareAlike 2.5.

Che cosa non è l'informatica.

Essere un informatico non vuol dire:

- saper scrivere un testo con Word;
- saper creare una presentazione con Powerpoint;
- saper usare la posta elettronica;
- saper navigare su internet;
- riuscire a scaricare da internet sul proprio telefonino la versione dal latino durante il compito in classe;
- essere esperti nell'uso di Excel;
- ...

Cioè non vuol dire:

- saper usare il calcolatore e gli strumenti informatici;
- avere ottenuto l'ECDL.

ECDL = European Computer Driving License

Patente Europea di Guida del Computer

- importante, perché nel mondo di oggi è indispensabile saper usare il calcolatore, così come per ora in Italia è (quasi) necessario saper guidare l'auto;
- attenzione, però: non confondere gli automobilisti con gl'ingegneri progettisti di automobili o con i fisici che studiano la termodinamica del motore a scoppio o i nuovi materiali per la costruzione dei diversi componenti di un'auto.

25/04/2007 16.08

E. Giovannetti - GalFer-07

3

Allora che cosa è l'informatica ?

- Una disciplina scientifica autonoma, "figlia ormai adulta" della matematica; su di essa si fonda la maggior parte delle applicazioni informatiche.
- L'informatica non è il saper usare il computer per il proprio lavoro o per i propri interessi personali:
 - fra un informatico e un esperto utente di computer c'è la stessa differenza esistente fra un bravo guidatore di auto e un ingegnere meccanico o un fisico;
 - un ingegnere meccanico è nella guida meno bravo di un pilota di Formula 1 o di un pilota di rally, ma le auto le progettano gl'ingegneri e i nuovi materiali li studiano e inventano i chimici e i fisici;
 - l'interazione fra ingegneri e piloti è tuttavia essenziale per il progresso della tecnica.

25/04/2007 16.08

E. Giovannetti - GalFer-07

4

Quali basi matematiche o di altro genere occorrono per iniziare lo studio dell'informatica ?

Quasi nessuna ! Ma sono indispensabili "capacità matematiche" e capacità di astrazione.

- La matematica dell'informatica è principalmente una matematica del discreto, e non del continuo come l'analisi.
- Lo studio dell'informatica può aiutare a comprendere concetti matematici, presentandoli sotto una diversa luce.
- Scrivere un programma corretto è difficile come risolvere un problema di geometria euclidea, in cui bisogna trovare la dimostrazione di una nuova proposizione (teorema).
- Per scrivere programmi corretti bisogna prima imparare definizioni, principi, regole, tecniche, ma poi non esistono ricette da applicare in modo meccanico !

25/04/2007 16.08

E. Giovannetti - GalFer-07

5

Sommario

- Alle basi della matematica: **che cosa sono i numeri ?**
La lingua dei polipi; i numeri nel calcolatore.
- Intermezzo linguistico: **che cosa è una grammatica ?**
Lingue naturali e linguaggi di programmazione.
- Paradigmi di programmazione e programmazione imperativa:
 - **variabili assegnabili e assegnazione;**
 - **istruzioni;**
 - **iterazione.**
- Alle basi dell'informatica: **che cos'è un algoritmo ?**
Esempio elementare.
- Problemi algoritmici e algoritmi.
- **Complessità computazionale.**
- Problemi difficili, problemi insolubili.
- Divagazioni filosofiche.

25/04/2007 16.08

E. Giovannetti - GalFer-07

6

Che cosa sono i numeri ?

25/04/2007 16.08

E. Giovannetti - GalFer-07

7

Che cosa sono i numeri ?

Studente (nel 2007): "Professore, $\frac{2}{3}$ lo devo scrivere come frazione o devo trasformarlo in numero ?"

Professore: "Argh !"

Ripasso: cominciamo dai numeri naturali.

L. Kronecker (1823-91):

"Die ganzen Zahlen hat der liebe Gott gemacht, alles andere ist Menschenwerk ."

(I numeri interi li ha fatti il Buon Dio, tutto il resto è opera dell'uomo)

25/04/2007 16.08

E. Giovannetti - GalFer-07

8

Base dieci e notazione posizionale

Anche gl'indiani lo pensavano all'incirca nello stesso modo:



ma lo scrivevano 23:

invece di scrivere **due** simboli del dieci e il simbolo del tre, scrivevano il simbolo del **due** a sinistra del simbolo del tre!

La posizione dei simboli diventa importante!

Numeri e rappresentazioni dei numeri

58 = 5 "mucchietti" da dieci + otto;
in italiano: **cinquantotto**, cioè 5dieci e 8

Perché lo *zero*?

5




= cinque rose o cinque gruppi di dieci rose?


cinque rose naturalmente!


Numeri e rappresentazioni dei numeri

58 = 5 mucchietti da dieci più otto;
 in italiano: **cinquantotto**, cioè 5dieci e 8

Perché lo zero?

5  = cinque cavalli o cinque gruppi di dieci cavalli?
 cinque cavalli, naturalmente!

50  = cinque gruppi di dieci cavalli
 + nessun (0) cavallo "fuori dai gruppi"

508  = cinque mandrie di cento cavalli,
 + nessun gruppo di 10 cavalli
 + otto cavalli

Come i polipi scrivono e pronunciano i numeri

I polipi non hanno dieci dita, bensì otto tentacoli; quindi
 raggruppano gli oggetti non per dieci, ma per otto
 (ricorda: in inglese polipo si dice octopus, dal greco = 8 piedi)
 Allora il numero che noi chiamiamo ventisei

••

loro lo pensano all'incirca così:

••••• ••••• ••••• ••
 ••••• ••••• ••••• ••

cioè 3 mucchietti da otto + 2,
 quindi lo scrivono 32,
 e lo pronunciano treottidue (o forse trottidue ...)

L'aritmetica dei polipi

uno	1	ottosette	17	...	
due	2	duotti	20	settottisette	77
tre	3	duottiuono	21	tento	100
quattro	4	duottidue	22	tille	1000
cinque	5	duottitre	23	...	
sei	6	duottiquattro	24		
sette	7	...			
otto	10	treotti	30		
ottouno	11	...			
ottodue	12	quattrotti	40		
ottotre	13	cinquotti	50		
...		ecc.			

25/04/2007 16.08

E. Giovannetti - GalFer-07

15

Traduzione dall'ottale al decimale

Come chiamiamo noi il numero
che i polipi scrivono 42 e chiamano quattrottidue?
Semplice:

$$4 \times 8 + 2 = 34$$

Nota Bene:

I piccoli dei polipi studiano a memoria delle tabelline con nomi e simboli diversi dai nostri.

Ad esempio la tabellina del 3 sarà:

$$3 \times 1 = 3; \quad 3 \times 2 = 6;$$

$$3 \times 3 = 11 \quad \text{cioè} \quad \text{tre per tre} = \text{ottuno};$$

$$3 \times 4 = 14 \quad \text{cioè} \quad \text{tre per quattro} = \text{ottoquattro}$$

...

$$3 \times 6 = 22 \quad \text{cioè} \quad \text{tre per sei} = \text{duotti-due}$$

25/04/2007 16.08

E. Giovannetti - GalFer-07

16

Traduzione dal decimale all'ottale

Come chiamano i polipi
il numero che noi chiamiamo 37?

$$37:8 = 4 \text{ col resto di } 5$$

lo chiamano 45 (quattrotticinque)
cioè $4 \times \text{otto} + 5$

Rappresentazione decimale

Dieci cifre designano i primi dieci numeri naturali
da zero a nove;
i numeri successivi sono rappresentati dalla scrittura:

$$c_k c_{k-1} \dots c_2 c_1 c_0 \text{ il cui significato è:}$$
$$c_k \times 10^k + c_{k-1} \times 10^{k-1} + \dots + c_2 \times 10^2 + c_1 \times 10^1 + c_0 \times 10^0$$

Rappresentazione ottale

otto cifre rappresentano i primi otto numeri naturali
da zero a sette; i numeri successivi sono rappresentati da:

$$c_k c_{k-1} \dots c_2 c_1 c_0$$

il cui significato è:

$$c_k \times 8^k + c_{k-1} \times 8^{k-1} + \dots + c_2 \times 8^2 + c_1 \times 8^1 + c_0 \times 8^0$$

Nota Bene: nel descrivere il significato del decimale e
dell'ottale noi umani scriviamo in decimale!

Addizione e sottrazione in ottale

ottale	decimale	ottale	decimale
32	26	23	19
<u>+43</u>	<u>+ 35</u>	<u>+ 45</u>	<u>+37</u>
75	61	70	56

ottale	decimale	ottale	decimale
23	19	43	35
<u>+47</u>	<u>+ 39</u>	<u>-25</u>	<u>-21</u>
72	58	16	14

25/04/2007 16.08

E. Giovannetti - GalFer-07

19

Il calendario dei polipi

- I polipi vivono nel 3727 (tretila-settetento-duotti-sette), non hanno avuto problemi del millennio, e aspettano con qualche moderata preoccupazione l'approssimarsi del quarto "tillennio" (corrispondente al nostro anno 2048).
- Un polipo-regista ha realizzato il film "4001 Odissea nello spazio"
- ...

25/04/2007 16.08

E. Giovannetti - GalFer-07

20

Frazioni e virgola decimale

Quanto fa $821 : 375$? La risposta è semplice: fa $\frac{821}{375}$

Le frazioni sono un nuovo tipo di numero rispetto ai naturali !

È che cosa è un "numero con la virgola (decimale)" ?

È un modo convenzionale di scrivere una somma di frazioni aventi come denominatori potenze successive di dieci.

$$0,3072 = \frac{3}{10} + \frac{0}{100} + \frac{7}{1000} + \frac{2}{10000}$$

Terminologia: frazione decimale.

È una frazione avente per denominatore una potenza di dieci.

$2/10$, $23/100$, $7/10000$, $5 (= 5/1)$ ecc. sono frazioni decimali

$100/7$ non è una frazione decimale

Frazioni e virgola decimale

Che cosa vuol dire che $7/4 = 1,75$?

Semplicemente che mangiare $7/4$ di torta è esattamente la stessa cosa che mangiare una torta e $7/10$ di torta e infine ancora $5/100$ di torta:

cioè la frazione $7/4$ è la stessa cosa che $1 + 7/10 + 5/100$

Scrivere le frazioni in questo modo ha il vantaggio che ad esempio si vede subito quante torte (pardon ..., quanti interi) ci sono nella frazione; inoltre, date due frazioni, possiamo immediatamente vedere quale di esse è più grande.

Frazioni non esprimibili per mezzo di frazioni decimali

Non tutte le frazioni sono somme **finite** di frazioni decimali:

$\frac{2}{3}$ non è esprimibile come somma finita di frazioni decimali,
ma solo come somma di infinite frazioni decimali:

$$6/10 + 6/100 + 6/1000 + 6/10000 + \dots$$

che scriviamo, nella notazione con la virgola:

0.66666

$$\frac{1}{7} = 0,142857142857142857 \dots$$

Le frazioni dei polipi

I polipi usano le frazioni ottali invece che decimali, e quindi usano una **virgola ottale**:

315 , 0473

significato:

$$3 \times 64 + 1 \times 8 + 5 + 0 \times (1/8) + 4 \times (1/64) + 7 \times (1/8^3) + 3 \times (1/8^4)$$

La frazione che noi scriviamo $1/8$ i polipi la scrivono $1/10$ (ma la pronunciano sempre "un ottavo"), cioè $0,1$.

La frazione che noi scriviamo $1/4$ ossia $25/100$ ossia $0,25$ in ottale si scrive $1/4$ ossia $2/10$ (due ottavi) ossia $0,2$

Numeri con la virgola ottale

In ottale:

$$1/5 = 0,146314631463 \dots$$

il numero con virgola decimale $0,2$
è un numero con virgola ottale periodico.

Il fatto che una frazione sia un numero con la virgola limitato oppure un numero con la virgola periodico dipende dalla base!

Rappresentazione in base 16: esadecimale

Se gli uomini avessero 16 dita, conterebbero in base sedici, e avrebbero inventato altri 6 segni (cioè cifre) per indicare i numeri da 10 a 15, ad es.:

0 1 2 3 4 5 6 7 8 9 ☺ ♀ Ω ☼ ☿
 10 11 ... 14 15

Allora ad es. il decimale 46,
 cioè due "sedicetti" + quattordici unità ($46 = 2 \times 16 + 14$),
 lo scriveremmo 2☼

Poiché la notazione esadecimale è usata solo nell'informatica, noi invece di inventarci nuovi simboli possiamo usare semplicemente le prime 6 lettere dell'alfabeto senza con ciò rischiare confusioni.

25/04/2007 16.08

E. Giovannetti - GalFer-07

27

decimale	esadecimale	decimale	esadec.	decimale	esadec.
...	...				
10	A	26	1A	255	FF
11	B	27	1B	256	100
12	C		
13	D	31	1F		
14	E	32	20		
15	F	33	21		
16	10		
17	11	48	30		
18	12		
19	13				
20	14				
21	15				
...	...				

25/04/2007 16.08

E. Giovannetti - GalFer-07

28

Lingue, linguaggi e grammatiche.

25/04/2007 16.08

E. Giovannetti - GalFer-07

29

Lingue, linguaggi e grammatiche

Da un punto di vista matematico:

- **Che cos'è una lingua?** È l'insieme infinito di tutte le possibili frasi corrette in quella lingua (~ una lingua coincide con il suo corpus ... potenziale).
- **Che cos'è una grammatica di una lingua?** È un insieme di regole formali che permette di **generare** - o, dualmente, di riconoscere - tutte le infinite frasi corrette di quella lingua.
- Correttezza grammaticale non implica aver senso.
Esempio: "**idee verdi senza colore dormono furiosamente**" è una frase italiana grammaticalmente corretta.
- Nelle lingue naturali la distinzione fra correttezza e scorrettezza grammaticale non è rigida.
- Nelle lingue naturali anche una frase scorretta può avere significato: "Io essere straniero".

25/04/2007 16.08

E. Giovannetti - GalFer-07

30

Grammatica di un frammento di italiano.

⟨frase⟩ → ⟨parte nominale⟩ ⟨parte verbale⟩.
⟨parte nominale⟩ → ⟨nome⟩ | ⟨nome⟩ ⟨relativa⟩
⟨parte verbale⟩ → ⟨verbointr⟩ | ⟨verbotr⟩ ⟨parte nominale⟩
⟨nome⟩ → ⟨nome proprio⟩ | ⟨articolo⟩ ⟨nome comune⟩
⟨relativa⟩ → **che** ⟨parte verbale⟩
⟨articolo⟩ → **il** | **un**
⟨nome comune⟩ → **cane** | **gatto**
⟨nome proprio⟩ → **Mario** | **Lucia**
⟨verbointr⟩ → **corre** | **scappa**
⟨verbotr⟩ → **insegue** | **raggiunge**

i simboli blu sono le *categorie sintattiche*;
i simboli rossi sono i *simboli terminali*.

25/04/2007 16.08

E. Giovannetti - GalFer-07

31

Riscriviamola, per ragioni di spazio, con una notazione abbreviata.

Usiamo **SN** (Sintagma Nominale) invece di ⟨parte nominale⟩,
SV (Sintagma Verbale) invece di ⟨parte verbale⟩, ecc.:

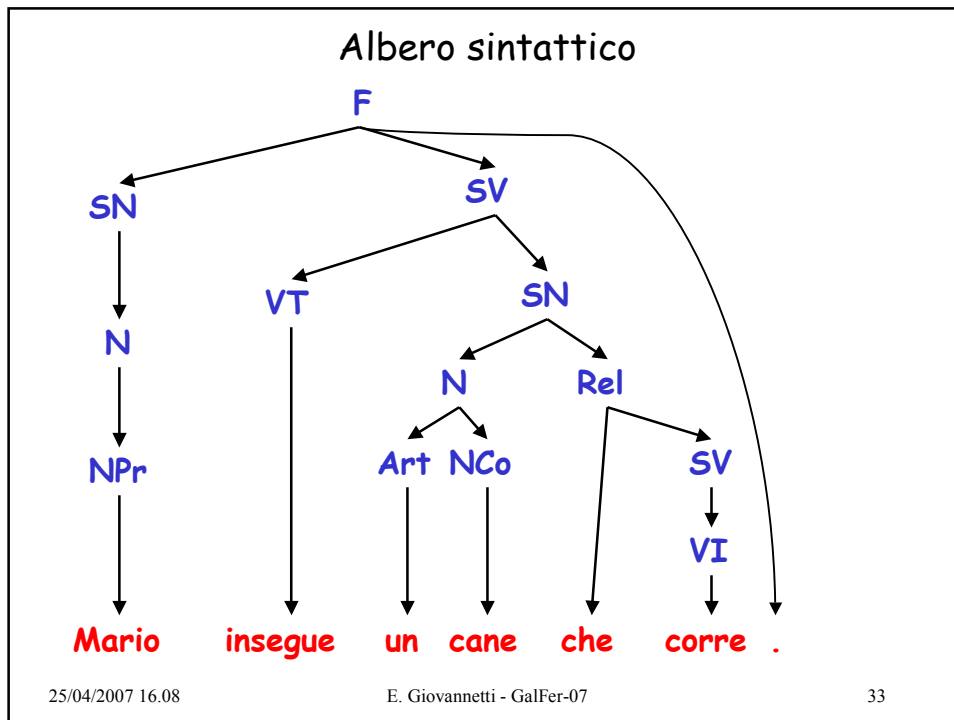
F → **SN SV**.
SN → **N** | **N Rel**
SV → **VI** | **VT SN**
N → **NPr** | **Art NCo**
Rel → **che SV**

Art → **il** | **un**
NCo → **cane** | **gatto**
NPr → **Mario** | **Lucia**
VI → **corre** | **scappa**
VT → **insegue** | **raggiunge**

25/04/2007 16.08

E. Giovannetti - GalFer-07

32



Esempi di frasi grammaticalmente corrette:

- Lucia corre.
- Lucia insegue il cane che insegue un gatto che scappa.
- un cane che insegue un gatto raggiunge Mario che raggiunge Lucia che insegue il cane.
- Mario insegue Mario.
- ...

Esempi di frasi grammaticalmente scorrette:

- il Mario corre.
- un cane raggiunge.
- corre il cane.
- il cane corre *(manca il punto finale)*
- ...

Un pezzo di una grammatica del TurboPascal

$\langle \text{programma} \rangle \rightarrow \langle \text{corpo-pr} \rangle . \mid \langle \text{intestazione} \rangle \langle \text{corpo-pr} \rangle .$
 $\langle \text{corpo-pr} \rangle \rightarrow \langle \text{dichiarazioni} \rangle \langle \text{istruzione-composta} \rangle$
 $\mid \langle \text{istruzione composta} \rangle$
 $\langle \text{istruzione-composta} \rangle \rightarrow \text{begin} \langle \text{sequenza-ist} \rangle \text{end}$
 $\langle \text{sequenza-ist} \rangle \rightarrow \langle \text{istruzione} \rangle \mid \langle \text{istruzione} \rangle ; \langle \text{sequenza-ist} \rangle$
 $\langle \text{istruzione} \rangle \rightarrow \epsilon \mid \langle \text{ist-semplce} \rangle \mid \langle \text{ist-strutturata} \rangle$
...

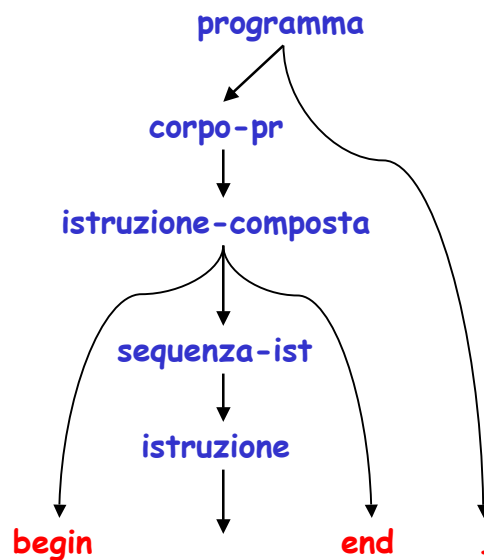
Nota: il simbolo ϵ denota la stringa vuota (corrispondente all'istruzione nulla).

25/04/2007 16.08

E. Giovannetti - GalFer-07

35

Il programma TurboPascal più corto del mondo.



25/04/2007 16.08

E. Giovannetti - GalFer-07

36

Il programma TurboPascal più corto del mondo.

```
begin  
end.
```

È un programma corretto, che può venire compilato ed eseguito, ma ... non è molto utile: non fa assolutamente nulla !

Paradigmi di Programmazione: Programmazione Imperativa.

La programmazione imperativa

È un modo di concepire e scrivere i programmi (o più in generale gli algoritmi) che "corrisponde" abbastanza al modo in cui funziona lo hardware della macchina:

- Un programma è costituito da una sequenza di **istruzioni** o **comandi** (per questo si chiama imperativa!).
- La macchina **esegue** il programma eseguendo un'istruzione dopo l'altra (alcune istruzioni hanno l'effetto di far eseguire un'istruzione invece di un'altra, oppure di far ripetere una sequenza di istruzioni, ecc.).
- La macchina ha uno **stato interno**, che è costituito dai contenuti delle celle di memoria, o **variabili**.
- Un'istruzione **può modificare tale stato**.
- Istruzioni di input/output permettono di comunicare con l'esterno, per ricevere i dati e fornire i risultati.
- Il paradigma imperativo non è l'unico!

25/04/2007 16.08

E. Giovannetti - GalFer-07

39

Programmazione imperativa: concetti fondamentali

variabile (o cella di memoria)

può essere pensata come un **contenitore** o **scatola**:

- dotata di un'etichetta esterna che è il suo indirizzo o nome,
- e contenente un valore, che è un ente astratto, come un numero, o un carattere, ecc.



25/04/2007 16.08

E. Giovannetti - GalFer-07

40

Programmazione imperativa: concetti fondamentali

variabile (o cella di memoria)

Nel programma Pascal (o Java, C, C#, ecc.) il nome della variabile non è il suo indirizzo numerico, bensì un nome simbolico dichiarato dal programmatore (o, come vedremo in seguito, un'espressione simbolica di altro genere).

La traduzione di tali nomi in indirizzi è fatta all'atto dell'esecuzione, senza intervento del programmatore Java (o C, ecc.).



25/04/2007 16.08

E. Giovannetti - GalFer-07

41

Programmazione imperativa: concetti fondamentali

ASSEGNAZIONE (assignment)

è un'istruzione (cioè un comando all'esecutore) che permette di cambiare il contenuto di una cella di memoria; esempi:

...

saldo := saldoIniziale

copia il contenuto della cella *saldoIniziale* nella cella *saldo*, cancellando il precedente contenuto di *saldo*;

saldo := saldo - prelievo

calcola la differenza fra il contenuto della cella *saldo* e quello della cella *prelievo*, e rimette il risultato nella cella *saldo*.

25/04/2007 16.08

E. Giovannetti - GalFer-07

42

Programmazione imperativa: assegnazione (cont.)

NOTA BENE

il simbolo "`:=`" NON indica l'uguaglianza matematica!

la scrittura

`a := b`

NON è un'espressione affermativa che *a* e *b* sono uguali;
è un comando o istruzione che copia in *a* il contenuto di *b*

Un simbolo più conveniente per l'assegnazione sarebbe:

`a ← b`

(l'informazione fluisce da destra verso sinistra)

Subito dopo l'esecuzione dell'istruzione i contenuti di *a* e di *b* saranno uguali, ma poi possono di nuovo variare indipendentemente.

25/04/2007 16.08

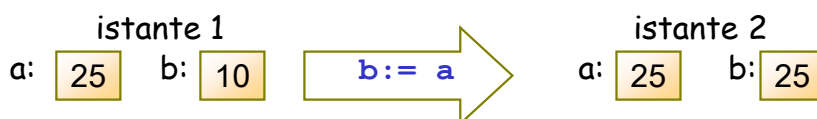
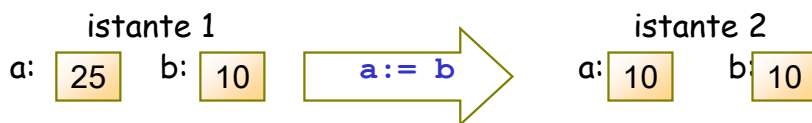
E. Giovannetti - GalFer-07

43

Programmazione imperativa: assegnazione (cont.)

NOTA BENE: dalla spiegazione precedente segue che l'operazione di assegnazione, a differenza dell'uguaglianza matematica, **NON è simmetrica**:

`a := b` NON è la stessa cosa che `b := a`



25/04/2007 16.08

E. Giovannetti - GalFer-07

44

L'assegnazione non è simmetrica (cont.)

`a := 3` è un'operazione perfettamente legale, che mette il valore 3 nel contenitore `a`, cancellando il valore precedente

`3 := a` è una scrittura priva di senso, perché 3 non è un contenitore! non si può mettere qualcosa nel numero 3!

`a := b+c` è un'operazione perfettamente legale, che mette in `a` la somma dei valori contenuti in `b` e in `c`

`b+c := a` è una scrittura priva di senso, perché l'*espressione* `b+c` non denota un contenitore, bensì un numero: la somma dei contenuti di `b` e `c` (si possono sommare i valori contenuti, NON i contenitori)

25/04/2007 16.08

E. Giovannetti - GalFer-07

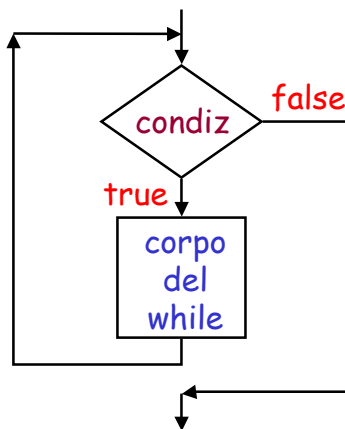
45

Iterazione: l'istruzione while.

Pascal
`while` condizione `do begin`
 corpo-del-while
`end`

C, C++, Java, C#, ecc.
`while` (condizione) {
 corpo-del-while
}

(unico esempio di diagramma di flusso)



25/04/2007 16.08

46

Alle basi dell'informatica. Che cos'è un algoritmo?

25/04/2007 16.08

E. Giovannetti - GalFer-07

47

Che cos'è un algoritmo ?

- *algoritmo non* è sinonimo di *diagramma di flusso*;
- un algoritmo è un procedimento (di calcolo) specificato per mezzo di regole precise e non ambigue, in modo da poter essere eseguito dall'uomo "meccanicamente", cioè applicando le regole senza pensare al significato;
- vi sono molti modi per descrivere un algoritmo; ad esempio l'algoritmo per la divisione in colonna imparato alle elementari è un algoritmo perfettamente specificato a parole;
- il metodo dei diagrammi di flusso viene ormai usato solo in casi molto particolari: in generale, per scrivere un programma, *non* serve fare prima il diagramma di flusso.
- a che cosa serve un algoritmo ?
a risolvere un ... problema algoritmico !

25/04/2007 16.08

E. Giovannetti - GalFer-07

48

Quando sono nati gli algoritmi ?

- algoritmi sono stati inventati ben prima della nascita dei calcolatori:
 - uno dei più antichi è l'algoritmo di Euclide per il MCD;



- un altro è il crivello di Eratostene per la generazione dei numeri primi;



25/04/2007 16.08

E. Giovannetti - GalFer-07

49

Algoritmi e ... logaritmi

- la parola *algoritmo* deriva dal nome del matematico persiano al-Khwarismi الخوارزمي (vissuto intorno all'anno 800), il cui libro "Calcoli con i numerali indiani" descriveva gli algoritmi di calcolo per le operazioni aritmetiche con il sistema di numerazione indiano, cioè quelli che ancora oggi studiamo nella scuola elementare;
- è un anagramma di *logaritmo*, parola che ha la stessa etimologia.



25/04/2007 16.08

E. Giovannetti - GalFer-07

50

L'algorithmica moderna.

- l'*algorithmica*, ovvero la scienza degli algoritmi, è fin dagli inizi dell'informatica una delle sue discipline centrali, ancora oggi attivo campo di ricerca.



Donald Knuth, padre dell'algorithmica moderna,
autore dell'opera in più volumi (3 pubblicati finora)
The Art of Computer Programming

25/04/2007 16.08

E. Giovannetti - GalFer-07

51

Algoritmi e macchine

Un algoritmo, essendo un metodo "meccanico", può essere eseguito da una macchina.

Si possono costruire macchine capaci di eseguire un particolare algoritmo, o un piccolo numero di algoritmi simili; ad esempio, una macchinetta calcolatrice è in grado di eseguire degli algoritmi corrispondenti alle quattro operazioni.

Un calcolatore (in inglese computer) è una macchina cosiddetta *universale*, cioè capace di eseguire - in linea di principio - qualunque algoritmo concepibile, specificato attraverso un *programma* (scritto in un linguaggio la cui forma dipende dalla macchina).

25/04/2007 16.08

E. Giovannetti - GalFer-07

52

Cos'è un problema algoritmico ? Esempi.

- **Problema del massimo di una sequenza di interi:** Data una sequenza (non vuota) di numeri interi, trovare il massimo di tale sequenza e l'indice (o posto) di un tale valore massimo nella sequenza.
- **Problema della ricerca in una sequenza ordinata:** Data una sequenza ordinata **S** di elementi e dato un elemento **x**, determinare se **x** compare nella sequenza o no, e se sì l'indice del posto in cui compare.
- **Problema del cammino minimo:** Data una mappa stradale che riporti località, strade e lunghezze dei percorsi, e data una coppia di località (raggiungibili l'una dall'altra), trovare un percorso di lunghezza minima fra di esse.
- **Problema della più lunga sottosequenza comune:** Date due sequenze di DNA, trovare la più lunga sottosequenza comune alle due sequenze (avendo definito in modo preciso le nozioni di sequenza di DNA e di sottosequenza).

25/04/2007 16.08

E. Giovannetti - GalFer-07

53

Problema algoritmico (o problema computazionale)

Idealmente consiste di:

- **precondizione:** una specifica precisa delle condizioni cui devono soddisfare i dati di partenza (input);
- **postcondizione:** una specifica precisa delle condizioni cui devono soddisfare i dati di arrivo (output), e in particolare della relazione che vi deve essere fra l'input e l'output;

NOTA BENE:

L'enunciazione di un problema algoritmico **NON** specifica **COME** ottenere l'output dall'input;

La descrizione precisa di **COME** ottenere l'output dall'input costituisce la **SOLUZIONE** del problema: è la descrizione di un algoritmo che risolve il problema.

25/04/2007 16.08

E. Giovannetti - GalFer-07

54

Efficienza di un algoritmo

Uno stesso problema può essere risolto da algoritmi diversi: due algoritmi diversi che risolvono uno stesso problema possono richiedere un numero di operazioni elementari e uno spazio di memoria molto diversi l'uno dall'altro.

Domande fondamentali:

un dato algoritmo quanto tempo ci mette, e di quanto spazio di memoria ha bisogno, per produrre la risposta per un input di una data dimensione ?

Oppure: qual è la massima dimensione dell'input per la quale l'algoritmo produce una risposta in un tempo accettabile e con un'occupazione di memoria accettabile ?

Dato un problema, si può inventare un algoritmo migliore di quelli esistenti per risolvere quel problema ?

Eccetera.

25/04/2007 16.08

E. Giovannetti - GalFer-07

55

Problemi algoritmici e algoritmi

- Un algoritmo è un metodo meccanico per risolvere un problema (o meglio un tipo di problema).
- Trovare (inventare) un algoritmo per risolvere (in maniera efficiente) un problema è un'attività non meccanica e non meccanizzabile ! è un'attività che richiede "intelligenza" !
- Non esiste un metodo meccanico che permetta, dato un problema, di trovare un algoritmo che risolva tale problema.
- "**Programming is mathematics made real**": programmare è un'attività che, in piccolo, può richiedere le stesse qualità che sono richieste a un grande matematico.
- Hilbert, a proposito di un suo ex-allievo: "Si è dato alla poesia. Aveva troppo poca fantasia per fare il matematico."
- Programmare è una forma di artigianato, con tecniche e regole, che richiede però una "disciplinata creatività".

25/04/2007 16.08

E. Giovannetti - GalFer-07

56

Esempio elementare: il massimo di una sequenza immessa da tastiera.

Esperimento.

Vi comunico lentamente una sequenza di numeri;
voi, senza usare né carta né matita né altri strumenti,
alla fine dovete dirmi qual è il massimo degli elementi.

...

Come avete fatto per dare la risposta giusta ?

Ad ogni passo:

- avete in memoria il massimo fino a quel momento;
- ricevete il nuovo numero;
- lo confrontate con il massimo:
- se è maggiore lo memorizzate come nuovo massimo dimenticando il precedente;
- altrimenti continuate a memorizzare il massimo

25/04/2007 16.08

E. Giovannetti - GalFer-07

57

Il massimo di una sequenza immessa da tastiera.

Servono due celle di memoria: una per tenere il "massimo fino a quel momento", e una per mettervi ogni volta il nuovo numero preso dalla tastiera. Chiamiamole **max** e **num**.

Corpo del while.

Ciò che facciamo ad ogni passo è:

```
num:=prendi_numero_da_tastiera;  
if num > max then max:= num;
```

Condizione del while.

Qual è la condizione per la quale eseguiamo il corpo ?
che la sequenza non sia finita:

```
while la_sequenza_non_è_finita do begin  
  num:=prendi_numero_da_tastiera;  
  if num > max then max:= num  
end
```

25/04/2007 16.08

E. Giovannetti - GalFer-07

58

Il massimo di una sequenza immessa da tastiera.

Inizializzazione.

Che cosa devo fare all'inizio, cioè prima di entrare nel ciclo ?
Il massimo è dato inizialmente dal primo numero immesso.
Allora:

```
max:= prendiNumeroDaTastiera;  
while la_sequenza_non_è_finita do begin  
    num:= prendiNumeroDaTastiera;  
    if num > max then max:= num  
end;
```

All'uscita dal ciclo, cioè alla fine della sequenza, la cella `max` contiene il massimo fino a quel momento, quindi contiene il massimo dell'intera sequenza. Se vogliamo visualizzarlo, basterà fare allora, ad esempio:

```
visualizza_su_schermo(max)
```

Il massimo di una sequenza immessa da tastiera.

Il programma finale

```
var num, max: integer;  
begin  
    writeln('immetti sequenza di int. non vuota terminata da CTRL/Z') ;  
    readln(max) ;  
    while not eof do begin  
        readln(num) ;  
        if num > max then max:= num  
    end;  
    writeln('massimo: ', max) ;  
    readln;  
end.
```

Nota: affinché il programma funzioni correttamente occorre che venga immesso almeno un numero.

Il massimo di una sequenza memorizzata in array

Sia **a** un array contenente **n** interi, indicati da **1** a **n**.

Partiamo dal programma per la sequenza da tastiera:

```
...  
begin  
  readln(max);  
  while not eof do begin  
    readln(num);  
    if num > max then max:= num  
  end;  
  writeln('massimo: ', max);  
  readln;  
end.
```

25/04/2007 16.08

E. Giovannetti - GalFer-07

61

Il massimo di una sequenza memorizzata in array

Sia **a** un array contenente **n** interi.

Partiamo dal programma per la sequenza da tastiera:

```
...  
begin  
  max:= a[1];  
  while not eof do begin  
    readln(num);  
    if num > max then max:= num  
  end;  
  writeln('massimo: ', max);  
  readln;  
end.
```

25/04/2007 16.08

E. Giovannetti - GalFer-07

62

Il massimo di una sequenza memorizzata in array

Sia **a** un array contenente **n** interi.

```
...
begin
  max:= a[1];
  for i:= 2 to n do begin
    readln(num);
    if num > max then max:= num
  end;
  writeln('massimo: ', max);
  readln;
end.
```

25/04/2007 16.08

E. Giovannetti - GalFer-07

63

Il massimo di una sequenza memorizzata in array

Sia **a** un array contenente **n** interi.

```
...
begin
  max:= a[1];
  for i:= 2 to n do begin
    num := a[i];
    if num > max then max:= num
  end;
  writeln('massimo: ', max);
  readln;
end.
```

25/04/2007 16.08

E. Giovannetti - GalFer-07

64

Il massimo di una sequenza memorizzata in array

Sia **a** un array contenente **n** interi.

```
...
begin
  max:= a[1];
  for i:= 2 to n do begin
    if a[i] > max then max:= a[i]
  end;
  writeln('massimo: ', max);
  readln;
end.
```