

**Università degli Studi della Calabria**

**Facoltà di Ingegneria**

**CORSO DI**

**INTRODUZIONE ALL' INFORMATICA**

**A.A 2002/2003**

Docente : Domenico Talia (talìa@deis.unical.it)

Esercitori: P. Trunfio, A. Congiusta

1

### **OBIETTIVI DEL CORSO**

- Illustrare i concetti fondamentali alla base degli attuali sistemi di elaborazione delle informazioni
- Introdurre l'uso degli elaboratori elettronici come strumenti per la soluzione di problemi
- Introdurre il concetto di sistema operativo
- Fornire conoscenze di base sulle reti informatiche
- Utilizzare strumenti di produttività individuale (elaborazione testi, gestione e analisi di dati)

2

## Programma del corso

Il modulo ha lo scopo di fornire allo studente una conoscenza di base della **struttura e del funzionamento di un personal computer** e delle reti informatiche, delle principali funzioni di un sistema operativo e dei **principali strumenti di produttività individuale** (gestione di testi, fogli elettronici) e della loro programmazione.

Inoltre, il modulo intende fornire le competenze necessarie per l'utilizzo di programmi di reperimento di informazioni e comunicazione su **Internet**, nonché di strumenti di editing e publishing su Web.

3

### ***Concetti introduttivi***

Hardware, software e Information Technology. L'hardware di un personal computer. I dispositivi di memoria. Tipi di software e loro funzioni. Reti informatiche. Cenni ai problemi di sicurezza nell'uso di sistemi informatici.

### ***Sistemi operativi***

Il sistema operativo Windows : caratteristiche e funzionalità. Organizzazione e struttura del file system. Menu e comandi.

### ***Strumenti di utilità personale I***

Sistemi di video scrittura. Word : caratteristiche e comandi principali.

### ***Strumenti di utilità personale II***

Fogli elettronici. Excel.

### ***Reti informatiche***

Concetti di Internet. Browser e Web. Ipertesti e navigazione. Indirizzamento di documenti. Gli URL. Ricerca di informazioni. Motori di ricerca. Creazione di documenti web. Posta elettronica e gestione di messaggi.

4

**Testi Consigliati:**

S. L. Nelson: *Microsoft Office 2000*, McGraw-Hill, 1999.

*oppure*

S. Rubini, M. De Marchi: *Moduli 1 e 2 - Concetti di base e Gestione dei file*, collana ECDL Apogeo, 2000.

S. Rubini: *Modulo 3 - elaborazione testi*, collana ECDL Apogeo, 2000.

P. Guccini: *Modulo 4 - foglio elettronico*, collana ECDL Apogeo, 2000.

G. Sitta, *Modulo 7 - reti informatiche*, collana ECDL Apogeo, 2000.

*consultazione*

D. Curtin et al., *Informatica di Base*, McGraw-Hill, 1999. 5

**Modalità di esame:** Prova pratica con possibilità del voto massimo e prova orale facoltativa con possibilità di miglioramento del voto della prova pratica.

**Materiale Didattico:** copia dei lucidi delle lezioni

<http://si.deis.unical.it/didattica/introinfo.html>

**Altri lucidi :** <http://www.dia.uniroma3.it/~cabibbo/fiji/>

(solo i capitoli 0,1,2)

Nota: parte dei lucidi sono stati realizzati dai Prof.ri Denti ed Omicini dell'Università di Bologna



### **Che cosa è l' INFORMATICA?**

#### **Esistono varie definizioni:**

- Scienza degli elaboratori elettronici (*Computer Science*)
- Scienza dell'informazione
- Informazione-automatica
- Scienza e tecnica dell'elaborazione dei dati e, genericamente, del trattamento automatico dell'informazione (Zingarelli).

L'informatica studia le caratteristiche dell'informazione e i modi di usarla, immagazzinarla, trasportarla e manipolarla in modo automatico.

#### **Elaboratore elettronico ("Computer"):**

è uno **strumento** per la rappresentazione, la memorizzazione e l'elaborazione delle informazioni.

•Che cos'è l'informatica?

*–Scienza del trattamento razionale, specialmente per mezzo di macchine automatiche, dell'informazione, considerata come supporto alla conoscenza umana e alla comunicazione (Academie Francaise).*

**Scienza della rappresentazione e dell'elaborazione dell'informazione**

•L'informatica ha due anime:

- **tecnologica**: i calcolatori elettronici e i sistemi che li utilizzano;
- **metodologica**: i metodi per la soluzione di problemi e la gestione delle informazioni.

9

**Hardware e Software**

La prima decomposizione di un calcolatore è relativa alle seguenti macro-componenti

**Hardware**

la struttura fisica del calcolatore, costituita da componenti elettronici ed elettromeccanici

**Software**

l'insieme dei programmi che consentono all'hardware di svolgere dei compiti utili

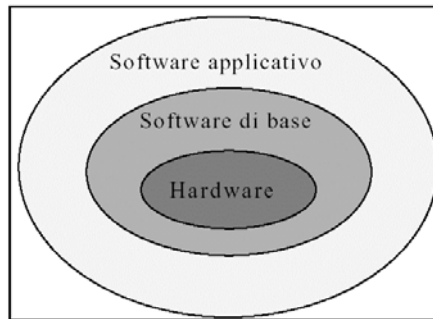
Il software comprende il **software di base** (tra cui il sistema operativo) e il **software applicativo**

10

## IL SOFTWARE

**Programmi** che vengono eseguiti dal sistema:

- Software di Base (es. Sistema Operativo)
- Software applicativo



11

### Software e macchine virtuali

Il software ha lo scopo di mostrare ai suoi utenti il calcolatore come una macchina virtuale più semplice da usare rispetto all'hardware che viene effettivamente utilizzato

**Macchina virtuale** : si intende una macchina che fisicamente non esiste nella realtà (ma funziona !)

Il **software di base** ha lo scopo di mostrare all'utente il calcolatore come una macchina virtuale più semplice da gestire e programmare rispetto all'hardware utilizzato

Il **software applicativo** mostra all'utente il calcolatore come una macchina virtuale che può essere utilizzata per la risoluzione di problemi.

12

Nell'organizzazione a livelli, l'hardware è **l'unica macchina reale**, mentre gli strati software corrispondono a macchine virtuali

Le operazioni (chiamate **istruzioni**) che l'hardware sa eseguire direttamente è chiamato **linguaggio macchina** del calcolatore ed è composto da istruzioni estremamente elementari, ma che il calcolatore sa eseguire in modo molto efficiente

### ***Problema di comunicazione tra uomo e macchina***

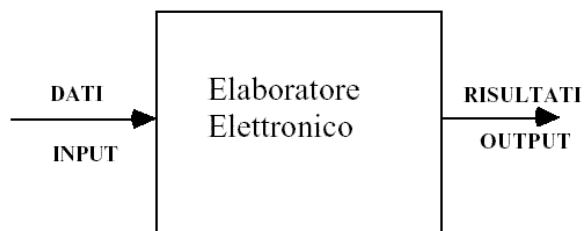
Le diverse macchine e i relativi insiemi di operazioni sono via via più astratti : più vicino alla logica dell'utente e più lontano dalla logica del calcolatore come dispositivo elettronico

Alla fine, comunque, l'unico responsabile dell'esecuzione del software è l'hardware disponibile

13

## **Programmazione**

È l'attività con cui si predispongono l'elaboratore ad eseguire un *particolare insieme di azioni* su *particolari dati*, allo scopo di *risolvere un problema*.



14

## I PROBLEMI DA RISOLVERE

I problemi che siamo interessati a risolvere sono di natura *molto varia*:

- Trovare il maggiore fra due numeri
- Dato un elenco di nomi e numeri di telefono, trovare il numero di una data persona
- Dati  $a$  e  $b$ , risolvere l'equazione  $ax+b=0$
- Stabilire se una parola precede alfabeticamente un'altra
- Identificare e prenotare aerei, treni, hotel...
- Ordinare un elenco di nomi

15

- Creare, modificare e alterare suoni
- Analizzare, riconoscere e modificare immagini .....
- Gestione delle aziende (private e pubbliche)

La descrizione del problema non indica direttamente (in genere) un modo per ottenere il risultato voluto

Differenza tra

*specifica di un **problema** e  
specifica del **processo di risoluzione***

16



## L'obiettivo fondamentale

Descrizione di un problema



Individuazione di un ALGORITMO

17

## ALGORITMO

***Sequenza finita di passi che risolve in un tempo finito un problema.***

*Esempi di algoritmi:*

1. Istruzioni di montaggio
2. Preparazione del caffè
3. Prelievo bancomat
4. Preparazione di un ricetta
5. Calcolo del massimo comun divisore tra due numeri interi

Proprietà di un algoritmo:

**Comprensibile, corretto ed efficiente**

18

La parola **algoritmo** deriva dal nome dell'autore del trattato dal cui titolo fu conosciuta la parola **algebra**.

Il persiano **Abu Ja'far Mohammed ibn Musa al-Khowarizmi** scrisse, circa nell'825, il trattato **Kitab al jabr w'al-muqabala**, che probabilmente significa **regole di trasposto e semplificazione**, e si riferisce alle regole per trasportare i termini da un membro all'altro di un'equazione e per semplificare.

Da **al jabr** derivò la parola **algebra** e dall'ultima parte del nome dell'autore, **Khowarizmi**, che indica la città di nascita, venne il termine **algoritmo**.

Originariamente era **agorismo** ma, per analogia con aritmetica, venne trasformato in **algoritmo**.

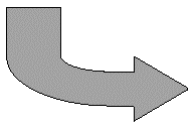
19

## ALGORITMO

Sequenza finita di passi che risolve in un tempo finito un problema.

### Codifica

Fase di scrittura di un algoritmo attraverso un insieme ordinato di frasi ("istruzioni"), scritte in un qualche **linguaggio di programmazione**, che specificano le **azioni** da compiere.



### Programma

**Testo** scritto in accordo alla **sintassi** e alla **semantica** di un linguaggio di programmazione

20

## **RISOLUZIONE DI PROBLEMI CON L'ELABORATORE ELETTRONICO**

### **Elaboratore elettronico**

- è uno strumento in grado di eseguire insiemi di *azioni* (“mosse”) elementari
- le azioni vengono eseguite su oggetti (*dati*) per produrre altri oggetti (*risultati*)
- l'esecuzione di azioni viene richiesta all'elaboratore attraverso *frasi* scritte in un qualche *linguaggio* (*istruzioni*)

21

## **RISOLUZIONE DI UN PROBLEMA**

Con questo termine si indica il processo che:

- dato un *problema*
- individuato un opportuno *metodo risolutivo* trasforma i dati iniziali nei corrispondenti risultati finali.

### **Alcune domande**

1) Si può risolvere un problema senza prima fissare un insieme di “azioni”, di “mosse elementari” possibili per l'elaboratore?

**No**, bisogna prima scendere al livello della macchina ( o in generale dell'esecutore), per conoscerne le caratteristiche, le mosse che sa eseguire ed il linguaggio che sa capire

22

# SINTASSI E SEMANTICA

**Sintassi** : l'insieme delle regole che consentono di scrivere parole e frasi riconoscibili come appartenenti ad un determinato linguaggio.  
Collegamento ordinato delle parole nel discorso.

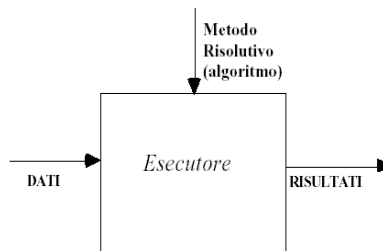
**Semantica** : la disciplina che studia il significato delle parole e delle frasi.

23

## Esecuzione

L'esecuzione delle azioni *nell'ordine specificato dall' algoritmo* consente di ottenere, a partire dai dati di ingresso, i risultati che risolvono il problema.

**Esecutore**: una macchina astratta capace di eseguire le azioni specificate dall'algoritmo.



24

## ALCUNE DOMANDE FONDAMENTALI

- **Quali istruzioni** esegue un elaboratore?
- **Quali problemi** può risolvere un elaboratore?
- **Esistono problemi che un elaboratore non può risolvere?**
- Che ruolo ha il linguaggio di programmazione?

## IL PROBLEMA DI FONDO

- Come si costruisce la soluzione a un problema?
- Qual è il giusto “punto di partenza” per pensare la soluzione a un problema?
- Quali metodologie e tecniche usare?

25

2) **È corretto** impostare la soluzione a partire da tali “mosse elementari” ?

- SI, per risolvere il problema *con efficienza*
- NO, se la macchina di partenza ha mosse di livello *troppo basso*



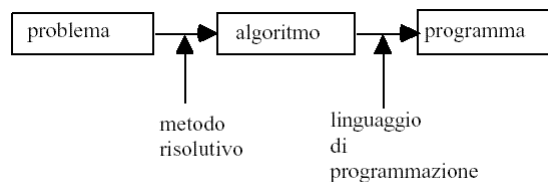
**Risalire a un più adeguato livello di *ASTRAZIONE*.**

*ASTRAZIONE*: processo di **aggregazione** di informazioni e dati e di **sintesi** di modelli concettuali che ne enucleano le proprietà **rilevanti** escludendo i dettagli inessenziali

26

## ALGORITMO E PROGRAMMA

- Ogni elaboratore è una macchina in grado di eseguire *azioni* elementari su *dati*
- L'esecuzione delle azioni elementari è richiesta all'elaboratore tramite comandi chiamati *istruzioni*
- Le istruzioni sono espresse attraverso *frasi* di un opportuno **linguaggio di programmazione**
- Un *programma* non è altro che la formulazione testuale di un algoritmo in un linguaggio di programmazione



27

### ALGORITMO: proprietà fondamentali

- 1) **Eseguibilità**: ogni azione deve essere *eseguibile* da parte dell'esecutore dell'algoritmo in un tempo finito
- 2) **Non-ambiguità**: ogni azione deve essere *univocamente interpretabile* dall'esecutore
- 3) **Finitezza**: il numero totale di azioni da eseguire, per ogni insieme di dati di ingresso, deve essere finito.

#### Quindi, l'algoritmo deve:

- essere *applicabile a qualsiasi insieme di dati di ingresso* appartenenti al **dominio di definizione dell'algoritmo**;
- essere costituito da operazioni appartenenti ad un determinato **insieme di operazioni fondamentali**
- essere costituito da **regole non ambigue**, cioè interpretabili in modo **univoco** qualunque sia l'esecutore (persona o "macchina") che le legge

#### Altre proprietà desiderabili

- generalità
- determinismo
- terminazione
- efficienza

28

Calcolo del Massimo Comun Divisore (MCD) fra due interi  
M ed N

### Algoritmo n° 1

- Calcola l'insieme A dei divisori di M
- Calcola l'insieme B dei divisori di N
- Calcola l'insieme C dei divisori comuni =  $A \cap B$
- Il risultato è il massimo dell'insieme C

29

### Algoritmo n° 2 (metodo di Euclide)

$$\text{MCD}(M,N) = \begin{cases} M \text{ (oppure N)} & \text{se } M=N \\ \text{MCD}(M-N, N) & \text{se } M>N \\ \text{MCD}(M, N-M) & \text{se } M<N \end{cases}$$

#### Algoritmo:

- Finché  $M \neq N$ :

se  $M > N$ , sostituisci a M il valore  $M' = M - N$

altrimenti sostituisci a N il valore  $N' = N - M$

- Il Massimo Comun Divisore è il valore finale ottenuto quando M e N diventano uguali.

30

## Algoritmo di Euclide

Dati due interi M e N

1. Dividi M per N e sia R il resto della divisione;
2. Se  $R=0$  allora termina e il Massimo Comun Divisore è N
3. Assegna a M il valore di N,  $M=N$ , a N il valore del resto,  $N=R$ , e torna al punto 1.

31

## RAPPRESENTAZIONE DELLE INFORMAZIONI

*Idea di fondo: Usare*

- *la presenza/assenza di carica elettrica*
- *il passaggio/non passaggio di corrente/luce.*

***BI***nary digi***T*** (cifra binaria): il ***BIT***

*Usiamo cioè' una rappresentazione binaria  
(a due valori) dell'informazione.*

32



## INFORMAZIONI COMPLESSE

*Con 1 bit rappresentiamo solo 2 diverse informazioni:*

si/no - on/off - 0/1

*Mettendo insieme piu' bit possiamo rappresentare piu' informazioni:*

00 - 01 - 10 - 11

*Con due bit si hanno quattro valori diversi.*

33

## INFORMAZIONI COMPLESSE

*In generale, con  $N$  bit, ognuno dei quali puo' assumere 2 valori, possiamo rappresentare  $2^N$  informazioni diverse*

viceversa:

*Per rappresentare  $M$  informazioni dobbiamo usare  $N$  bit, in modo che:  $2^N \geq M$*

34

## ESEMPIO:

*Per rappresentare 57 informazioni diverse dobbiamo usare gruppi di almeno 6 bit. Infatti:*

$$2^6 = 64 > 57$$

*Cioe' un gruppo di 6 bit puo' assumere 64 configurazioni diverse:*

*000000 / 000001 / 000010 ... / 111110 / 111111*

35

## IL BYTE:

*In informatica ha assunto particolare importanza il concetto di **byte**:*

$$1 \text{ byte} = 8 \text{ bit} = 2^8 = 256 \text{ valori diversi}$$

*Il byte e' usato come unita' di misura per indicare le dimensioni della memoria, la velocita' di trasmissione, la potenza di un elaboratore*

*Usando sequenze di byte (e quindi di bit) si possono rappresentare caratteri, numeri immagini, suoni.*

36

## IL SISTEMA DECIMALE

- 10 cifre di base: 0, 1, 2, ..., 9
- **Notazione posizionale:** la posizione di una cifra in un numero indica il suo **peso** in potenze di **10**. I pesi sono:

– unita'                    =  $10^0 = 1$                     (posiz. 0-esima)

– decine                    =  $10^1 = 10$                     (posiz. 1-esima)

– centinaia                =  $10^2 = 100$                     (posiz. 2-esima)

– migliaia                 =  $10^3 = 1000$                     (posiz. 3-esima)

– ...                        ..                    ..                    ..                    ..                    ..                    ..

37

## ES. DI NUMERO RAPPR. IN NOTAZIONE DECIMALE

*Il numerale 2304 in notazione decimale (o in base 10) rappresenta la quantità':*

$$2304 = 2 \cdot 10^3 + 3 \cdot 10^2 + 0 \cdot 10^1 + 4 \cdot 10^0 =$$

$$2000 + 300 + 0 + 4 = 2304 \text{ (**numero**)}$$

*Nota: numero e numerale qui coincidono, perché il sistema decimale è quello adottato come sistema di riferimento.*

38

# IL SISTEMA BINARIO

- *2 Cifre di base: 0 e 1.*
- **Notazione posizionale:** la posizione di una cifra in un numero binario indica il suo **peso** in potenze di **2**.  
I pesi sono:
  - $2^0 = 1$  (posiz. 0-esima)
  - $2^1 = 2$  (posiz. 1-esima)
  - $2^2 = 4$  (posiz. 2-esima)
  - $2^3=8; 2^4=16; 2^5=32; 2^6=64; 2^7=128; 2^8=256; 2^9=512;$   
 $2^{10} = 1024; 2^{11}=2048, 2^{12}=4096;...$

39

## ES. DI NUMERO RAPP. IN NOTAZIONE BINARIA

*Il numerale 10100101 in notazione binaria (o in base 2) rappresenta la quantità:*

*10100101*

$$1*2^7+0*2^6+1*2^5+0*2^4+0*2^3+1*2^2+0*2^1+1*2^0$$

$$128 + 0 + 32 + 0 + 0 + 4 + 0 + 1 =$$

*165 (numero)*

40

## IL NUMERO PIU' GRANDE RAPPRESENTABILE CON N CIFRE

- *Sist. Decimale* =  $99\dots99 = 10^N - 1$
- *Sist. Binario* =  $11\dots11 = 2^N - 1$
- **Esempio:**  $11111111$  (8 bit binari) =  $2^8 - 1 = 255$ . Per rappresentare il n. 256 ci vuole un bit in piu':  $100000000 = 1 * 2^8 = 256$ .

41

## QUINDI:

Fissate quante cifre (bit) sono usate per rappresentare i numeri, si fissa anche il numero piu' grande che si puo' rappresentare:

- con 16 bit:  $2^{16} - 1 = 65.535$
- con 32 bit:  $2^{32} - 1 = 4.294.967.295$
- con 64 bit:  $2^{64} - 1 = \text{circa } 1,84 * 10^{19}$

42