

Laboratorio di Informatica

Evoluzione del calcolatore e fondamenti della programmazione Lezione 1

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

1

Laboratorio di Informatica
1. Introduzione

Informatica

L'informatica è la disciplina che fonda su basi scientifiche diverse aree:

- il disegno e la progettazione di computer
- la programmazione
 - Includendo l'implementazione di nuovi strumenti per rendere il calcolatore fruibile ai non addetti ai lavori
- il trattamento dell'informazione e la gestione dei dati
 - Includendo l'individuazione di nuovi ambiti applicativi
- la soluzione algoritmica di problemi
- lo studio del processo algoritmico in sé

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

2

Laboratorio di Informatica
1. Introduzione

All'origine

- I calcolatori
 - nati in risposta all'esigenza di eseguire *meccanicamente* operazioni ripetitive;
- Gli algoritmi
 - nati in risposta all'esigenza di definire procedure *meccaniche* per la soluzione di problemi

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

3

Laboratorio di Informatica
1. Introduzione

I Calcolatori

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

4

Laboratorio di Informatica
1. Introduzione

Storia del calcolatore

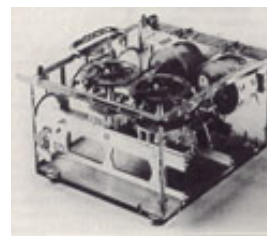
- I primi tentativi di automatizzare il calcolo risalgono al '500
 - 1623, Schickard: il primo modello di calcolatore per sommare e sottrarre in modo automatico, moltiplicare e dividere in modo semi-automaticamente
 - Fu costruita ma se ne perse traccia e non influenzò i prototipi successivi

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

5

Laboratorio di Informatica
1. Introduzione

Pascal (1623-1650)



La Pascalina, la prima macchina di calcolo, meno potente ma funzionante

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

6

Laboratorio di Informatica
1. Introduzione

Leibniz (1646-1716)

“Non è degno di uomini eccellenti perdere ore come schiavi a faticare su calcoli che potrebbero essere affidati a chiunque se venissero usate le macchine”

- Macchine basate sul modello della *ruota di Leibniz* sono state sostanzialmente in uso fino a tempi recenti
- Automatizzava completamente le 4 operazioni aritmetiche

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

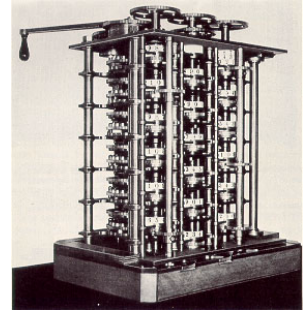
7

Laboratorio di Informatica
1. Introduzione

Babbage (1791-1871)

Inventa le macchine a vapore:

- La macchina analitica e la macchina alle differenze
- Questa veniva azionata da cartoncini perforati
- Un'idea di programma
- Ada Lovelace



8

Laboratorio di Informatica
1. Introduzione

Jaquard – 1801

- Il telaio per tessitura, controllato da schede perforate
- Tipi diversi di tessitura richiedevano solo di cambiare le schede

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

9

Laboratorio di Informatica
1. Introduzione

Mark I

- Una macchina di calcolo elettro-meccanica, costruita da Howard Aiken alla Harvard University in collaborazione con IBM



AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

10

Laboratorio di Informatica
1. Introduzione

Calcolatore elettronico – 1940

- ABC (Atanasoff Berry Computer)
 - Special purpose
 - Costruita ma poco usata
 - Valvole
 - Influenzò molto gli esperimenti successivi
 - in particolare Mauchly che poi costruisce l'ENIAC
- Colossus
 - Costruito in Inghilterra per rompere i codici crittografati dai tedeschi nella II guerra mondiale

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

11

Laboratorio di Informatica
1. Introduzione

ENIAC (1940 – 1946)

- Il primo calcolatore elettronico *general purpose* di John Mauchly e J. Presper Eckert
- Realizzato alla Moore School di Penn. State
- Costituito da
 - 8.000 valvole
 - 70.000 resistori
 - 10.000 condensatori
 - 6.000 interruttori
- Era lungo 30 metri, consumava 140 KW

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

12

Laboratorio di Informatica
1. Introduzione

John Von Neumann (1903-1957)

- Matematico. Ha dato contributi fondamentali in tutti i campi di cui si è occupato:
 - Logica
 - Matematica
 - Meccanica quantistica
 - Informatica
 - Teoria di Giochi
 - Economia
 - Armamenti - Politica
 - Voleva esser ricordato per gli studi in Cibernetica
- <http://www.dipmat.unipg.it/~bartocci/odifr1.html>

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

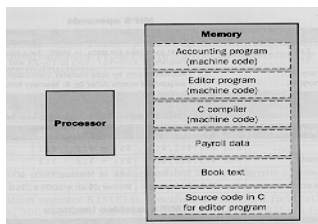
13

Laboratorio di Informatica
1. Introduzione**Von Neumann e ENIAC**

- Nel 1944 Von Neumann si aggiunge al gruppo ENIAC
- Insieme studiano metodi più efficienti per programmare un calcolatore
- Nasce l'idea di calcolatore a programma memorizzato (Macchina di Von Neumann) e viene dato avvio al progetto EDVAC
- Progetta la macchina dell'Institute of Advanced Studies di Princeton (lo IAS)

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

14

Laboratorio di Informatica
1. Introduzione**La macchina di von Neumann**AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

15

Laboratorio di Informatica
1. Introduzione**La macchina di Von Neumann**

- La memoria contiene i dati e il programma
- Unità di processo
- Strutture di controllo per l'esecuzione sequenziale delle istruzioni
- Grande influenza sui linguaggi di programmazione

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

16

Laboratorio di Informatica
1. Introduzione**EDVAC - EDSAC**

- Il progetto EDVAC viene portato a termine nel 1952
- Il primo calcolatore a programma memorizzato (EDSAC – electronic delay storage automatic calculator) viene realizzato all'Università di Manchester da M. Wilkes nel 1949

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

17

Laboratorio di Informatica
1. Introduzione**Il calcolatore moderno**

- I generazione, fino al 1959
 - valvole
 - alto consumo, surriscaldamento
- II generazione, 1959 - 1964
 - i transistor
 - macchine più piccole, più affidabili e più economiche

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

18

Laboratorio di Informatica
1. Introduzione

I calcolatori commerciali

- Tutti i prototipi sinora discussi sono realizzati in ambito accademico
- Il primo calcolatore realizzato per scopi commerciali è l'UNIVAC I, costruito da Eckert-Mauchly
 - Venduto a 1M\$
 - 48 esemplari venduti

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

19

Laboratorio di Informatica
1. Introduzione*I calcolatori commerciali*

- Nel 1964 IBM presenta l'elaboratore System/360, costo medio 500.000\$
- Nel 1965 DEC presenta il primo minicomputer il PDP-8, costo medio 20.000\$
- Nel 1977 S. Jobs e S. Wozniak propongono il primo personal computer Apple II, costo medio 2000\$

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

20

Laboratorio di Informatica
1. Introduzione*Il calcolatore moderno*

- III generazione, 1965 – 1970
 - i circuiti integrati e miniaturizzati con molti transistor
- IV generazione, i microcomputer
 - circuiti LSI e VLSI
 - la memoria usa la tecnologia dei semiconduttori

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

21

Laboratorio di Informatica
1. Introduzione*Le generazioni di calcolatori*

Generation	Dates	Technology	Principal new product
1	1950-1959	Vacuum tubes	Commercial electronic computer
2	1960-1968	Transistors	Cheaper computers
3	1969-1977	Integrated circuit	Minicomputer
4	1978-?	LSI and VLSI	Personal computers and workstations

FIGURE 1.29 Computer generations are usually determined by the change in dominant implementation technology. Typically, each generation offers the opportunity to create a new class of computers and to create new computer companies.

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

22

Laboratorio di Informatica
1. Introduzione*Le generazioni di calcolatori*

Year	Name	Size (cu. ft.)	Power (watts)	Performance (adds/sec)	Memory (KB)	Price	Price/performance vs. UNIVAC
1951	UNIVAC I	1000	124,500	1,900	48	\$1,000,000	1
1964	IBM S/360 model 50	60	10,000	500,000	64	\$1,000,000	263
1965	PDP-8	8	500	330,000	4	\$16,000	10,855
1976	Cray-1	58	60,000	166,000,000	32,768	\$4,000,000	21,842
1981	IBM PC	1	150	240,000	256	\$3,000	42,105
1991	HP 9000/ model 750	2	500	50,000,000	16,384	\$7,400	3,556,188
1996	Intel PPro PC (200 MHz)	2	500	400,000,000	16,384	\$4,400	47,846,890

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

23

Laboratorio di Informatica
1. Introduzione*Gli algoritmi*AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

24

Laboratorio di Informatica
1. Introduzione

La soluzione algoritmica di problemi

- Algoritmo, il concetto fondamentale e centrale dell'informatica
 - da AL-KHOWARIZMI (825 dc). Una procedura per risolvere un problema matematico in un numero finito di passi che implicano frequenti ripetizioni di un'operazione.
 - In senso lato, una procedura che eseguita a passo a passo risolve un problema.
 - Dal dizionario Webster

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

25

Laboratorio di Informatica
1. Introduzione

Esempi

- Algoritmi (o procedure):
 - per calcolare il Massimo Comun Divisore
 - per costruire modellini di aerei (espressi nei fogli di istruzioni)
 - per azionare la lavatrice
 - per suonare una melodia al piano (espressa in un insieme di simboli negli spartiti)

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

26

Laboratorio di Informatica
1. Introduzione

Algoritmo di Euclide

- Dati due numeri interi e positivi m e n , calcola il più grande intero che li divide

1. dividere m per n , e sia r il resto della divisione (con $0 \leq r < n$)
2. se $r = 0$ allora la risposta è n e STOP
3. porre $m \leftarrow n$ e $n \leftarrow r$ e ripetere il passo 1.

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

27

Laboratorio di Informatica
1. Introduzione

Verifica empirica

- Funziona davvero? Il ciclo dei 3 passi termina sempre?
- Proviamo con $m \leftarrow 119$ e $n \leftarrow 544$:

1. $119 / 544$ dà resto $r = 119$
2. se $r \neq 0$ allora
3. poniamo $m \leftarrow 544$ e $n \leftarrow 119$. Torniamo al passo 1.

1. $544 / 119$ dà resto $r = 68$
2. se $r \neq 0$ allora
3. poniamo $m \leftarrow 119$ e $n \leftarrow 68$. Torniamo al passo 1.

1. $119 / 68$ dà resto $r = 51$
2. se $r \neq 0$ allora
3. poniamo $m \leftarrow 68$ e $n \leftarrow 51$. Torniamo al passo 1.

1. $68 / 51$ dà resto $r = 17$
2. se $r \neq 0$ allora
3. poniamo $m \leftarrow 51$ e $n \leftarrow 17$. Torniamo al passo 1.

1. $51 / 17$ dà resto $r = 0$
2. se $r = 0$ allora il **MCD** = $n = 17$

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

28

Laboratorio di Informatica
1. Introduzione

Algoritmo vs procedura di calcolo

- Parliamo di **algoritmi** solo se definiscono un insieme **finito** di istruzioni che danno luogo a una sequenza **finita** di operazioni
- Il processo di ripetizione di cicli **deve terminare**
 - Nell'algoritmo di Euclide, la sequenza dei resti r
 - sequenza di numeri interi positivi decrescenti
- Non tutte le procedure soddisfano questo requisito
 - Si parla allora di procedure di calcolo

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

29

Laboratorio di Informatica
1. Introduzione

Trovare algoritmi

- La ricerca di algoritmi è stata una grande parte del lavoro dei matematici nei secoli
- Gli algoritmi rendono il lavoro più semplice
- La loro esecuzione non richiede la comprensione dei principi su cui si fonda
- Il computer è un esecutore di algoritmi

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

30

Laboratorio di Informatica
1. Introduzione

I linguaggi di programmazione

- L'algoritmo va rappresentato in un sistema formale per essere comunicato alla macchina che lo esegue
- I linguaggi di programmazione
- Una varietà di approcci per affrontare il processo della programmazione

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

31

Laboratorio di Informatica
1. Introduzione

Sintassi

- Sistema di regole formali che definiscono il linguaggio
- Consente di stabilire se un'istruzione è ben formata
 - È corretto scrivere: **se $x + n > m$ allora STOP ?**
- Facilitano il compito al programmatore
- I programmi devono essere tradotti nel linguaggio nativo della macchina

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

32

Laboratorio di Informatica
1. Introduzione

Diversi livelli di espressività

- In un linguaggio ad alto livello, PASCAL
 - **semiperimetro := base + altezza;**
- In un linguaggio assembly, simbolico del linguaggio macchina,
 - **LOAD x**
 - **ADD y**
 - **STORE z**

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

33

Laboratorio di Informatica
1. Introduzione

Linguaggi di basso livello

- In un linguaggio macchina, ogni istruzione è una sequenza di cifre binarie
 - Totalmente illeggibile per l'uomo
 - Perfettamente non ambiguo per la macchina

```
0010000000000100
0100000000000101
0011000000000110
```

N.B. traduzione delle istruzioni precedenti

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

34

Laboratorio di Informatica
1. Introduzione

Traduzione dei linguaggi

- Il concetto di traduzione dei linguaggi ha permesso l'evoluzione dei linguaggi verso sistemi simbolici più espressivi e più facilmente manipolabili dai programmatori
- Il programmatore scrive un programma in un linguaggio ad alto livello senza preoccuparsi della macchina che esegue il programma
- Il compilatore viene predisposto per una macchina specifica e poi traduce tutti i programmi scritti in uno specifico linguaggio ad alto livello

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

35

Laboratorio di Informatica
1. Introduzione

Programma compilatore

- Il compilatore traduce istruzioni scritte in un linguaggio ad alto livello, definite da regole sintattiche precise
- Necessità di strumenti formali per la descrizione delle grammatiche (**tavole sintattiche, grammatiche, BNF**)
- Programma **sorgente**: il programma ad alto livello tradotto dal compilatore
- Programma **oggetto**: il risultato della traduzione del compilatore

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

36

Laboratorio di Informatica
1. Introduzione

Influenza del modello sui linguaggi

- Concetto di istruzione
 - l'unità di base del programma, memorizzate in successive celle di memoria
- Concetto di sequenzialità e iterazione
 - In diversi costrutti dei linguaggi e in tutto il processo di esecuzione
- Concetto di variabile
 - Le celle di memoria contenenti i valori da manipolare

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

37

Laboratorio di Informatica
1. Introduzione

È sorprendente che il computer di Von Neumann sia rimasto così a lungo il paradigma fondamentale dell'architettura dei calcolatori.

Ma dato il fatto, non è sorprendente che i linguaggi imperativi siano i principali oggetti di studio e sviluppo.

Perché come Backus ha sottolineato i linguaggi imperativi hanno solide radici nell'architettura della macchina di Von Neumann e ne sono l'immagine.

Horowitz (Fundamentals of Programming Languages, 1983)

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

38

Laboratorio di Informatica
1. Introduzione

Evoluzione dei linguaggi

- Anni '50: i primi linguaggi ad alto livello
 - **FORTRAN**, introduce il concetto di sottoprogrammi che operano su dati comuni
 - **ALGOL**, introduce il concetto di struttura dei programmi e di procedure ricorsive, ovvero che richiamano se stesse
 - **COBOL**, introduce il concetto di FILE e di descrizione dei dati

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

39

Laboratorio di Informatica
1. Introduzione

Evoluzione dei linguaggi

- Anni '60: i linguaggi orientati al problema
 - **LISP**, introduce uniformità tra dati e programmi e un paradigma di programmazione basato sul concetto di funzione
 - **APL**, linguaggio matematico, ricco di notazioni e operatori per operare su strutture come vettori e matrici
 - **SNOBOL**, offre strumenti utili per la manipolazione di sequenze di caratteri

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

40

Laboratorio di Informatica
1. Introduzione

Evoluzione dei linguaggi

- Anni '70: metodologia di programmazione
 - **PASCAL**, ha lo scopo di insegnare la programmazione strutturata, una possibile risposta alla necessità di programmare con un metodo. Fa seguito **Modula-2** che introduce il concetto di modulo.
 - **C**, linguaggio ad alto livello con visibilità e accesso alla macchina
 - **Prolog**, linguaggio basato sulla logica, non convenzionale diventato di nicchia

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

41

Laboratorio di Informatica
1. Introduzione

Evoluzione dei linguaggi

- Anni '80: programmazione in grande, esigenza di modularità e di astrazione
 - La programmazione a oggetti. Una classe definisce un insieme di oggetti e le procedure per manipolarli
 - **SmallTalk**, **Objective-C**, **C++**

AA 2000/01
© Alberti, Bruschi, Ferrari, Provetti, Rosti

42

Laboratorio di Informatica
1. Introduzione

Teoria degli algoritmi

- Gli algoritmi risolvono un problema
 - Possibilmente sono corretti ...
 - Dimostrazioni di **correttezza** degli algoritmi vs verifica empirica
 - **Complessità** degli algoritmi
 - Se gli algoritmi sono corretti, sono *buoni*?
 - Criteri di bontà: efficienza nell'uso delle risorse sia di tempo di calcolo, sia di occupazione di memoria
 - Problemi intrinsecamente difficili