

Introduzione al corso di Elementi di Programmazione

Prof. Salvatore Venticinque

Informazioni utili

Prof. Salvatore Venticinque

Prof. Pietro Ferrara

- Contatti:

Dipartimento di Ingegneria

mailto: salvatore.venticinque@unicampania.it

piteferrara@gmail.com

<https://elearning.unicampania.it>

<http://www.ingegneria.unicampania.it>

Per le comunicazioni: forum del corso (collegato a email ufficiale e rss-feeds)

- Ricevimento studenti:

Venerdì 11:30-13:00 Prof. Salvatore Venticinque

Martedì 11:30-13:00

Orario

- LUN 16:00-18:00 (Esercitazioni)
- MAR 9:00-11:00
- GIOV 14:00-16:00
- VEN 9:00-11:00

Articolazione del corso

Elementi di Programmazione (9CFU) - ING-INF/05

- Argomenti teorici di base
 - oggetto delle lezioni in aula
 - verificati tramite quesiti ed esercizi della prova orale

- Programmazione in linguaggio C
 - oggetto delle lezioni e delle esercitazioni in aula
 - verificata tramite problemi della prova scritta

Modalità d'esame

- **Prova scritta.**

- Composta di una parte applicativa
- Obbligatoria per il superamento dell'esame

- **Prova orale.**

Libri di testo

Testo di riferimento:

Bellini, Guidi. Linguaggio C. Mc-Graw-Hill,

Altri testi consigliati:

- Kernighan, Ritchie - "Il linguaggio C", Pearson
- Pasquale Foggia, Mario Vento - "Algoritmi e strutture dati", McGraw Hill

Programma

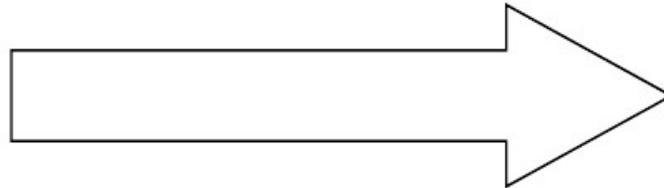
- Lezioni frontali (9 crediti, 72 ore):
 - Algoritmi e Programmazione Strutturata
 - Programmazione codifica in linguaggio C
- Esercitazioni in aula con portatile personale
 - Progettazione di Algoritmi
 - Codifica di Algoritmi in Linguaggio C
 - Progettazione e Codifica Algoritmi di Ordinamento e Ricerca
 - Uso di funzioni e procedure e gestione dei file
 - Utilizzo degli ambienti di sviluppo

Il programma completo sarà disponibile sul sito del corso.

Cosa Impariamo in questo Corso



Costruzione di
un programma



- In C



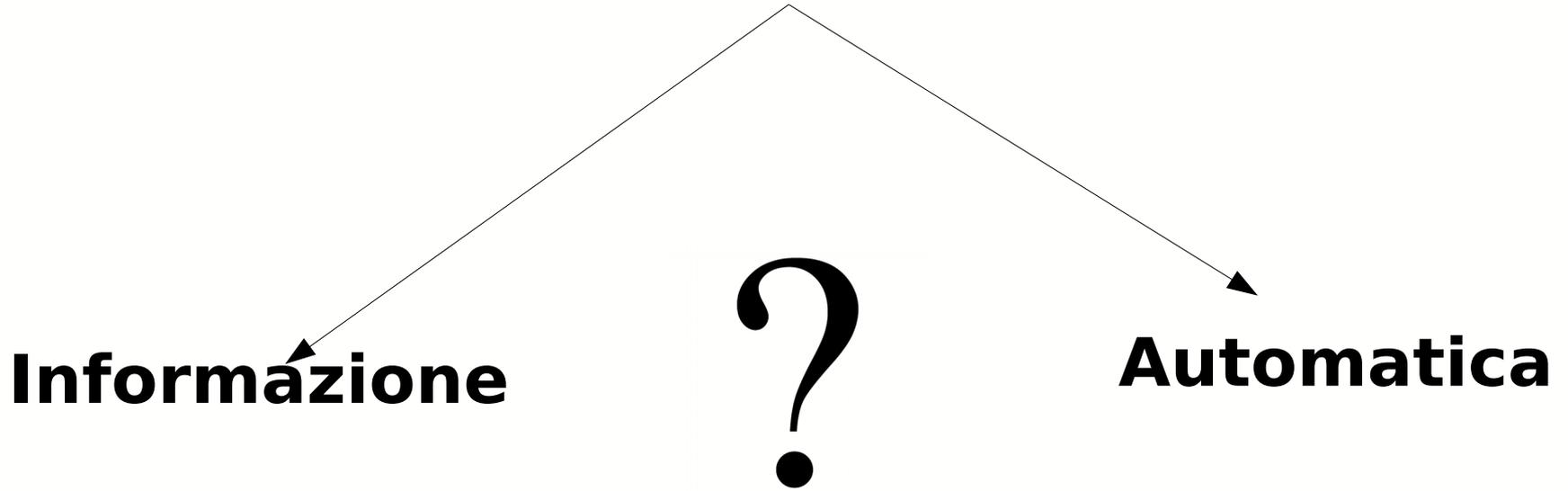
Informatica: etimologia

Informatica

Informazione

?

Automatica



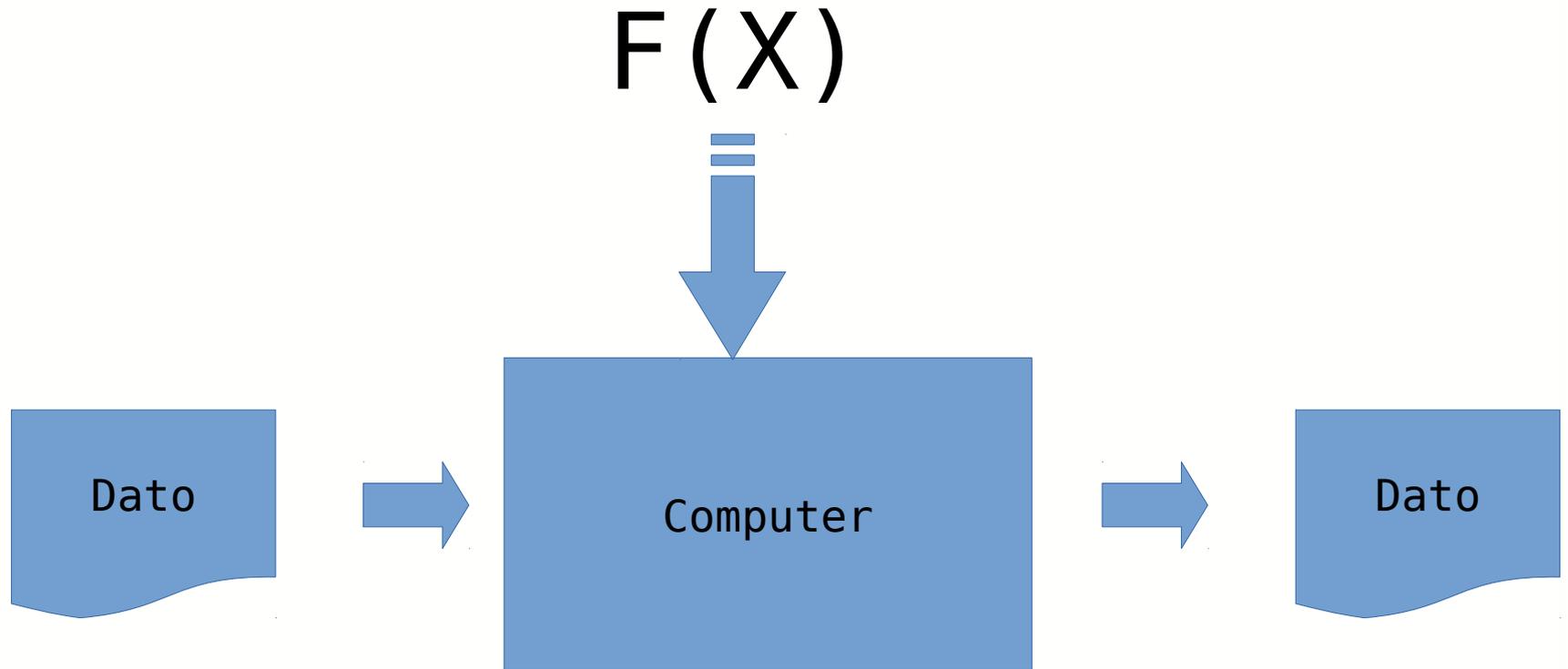
Elaborazione

Elaborazione automatica dell'informazione

Cosa serve per l'elaborazione automatica?

***un sistema digitale a programma registrabile per l'elaborazione
automatica dell'informazione***

L'Elaboratore



Informatica

- Definizione

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

- Definizione Association of Computing Machinery(ACM)

Studio sistematico degli algoritmi(sequenze precise di operazioni comprensibili e perciò eseguibili da uno strumento automatico) che descrivono e trasformano l'informazione: la loro teoria, analisi, progetto, efficienza, realizzazione e applicazione

Modello di Sistema a Programma Registrabile

Programma
(funzione di trasformazione)



Ingressi Principali:

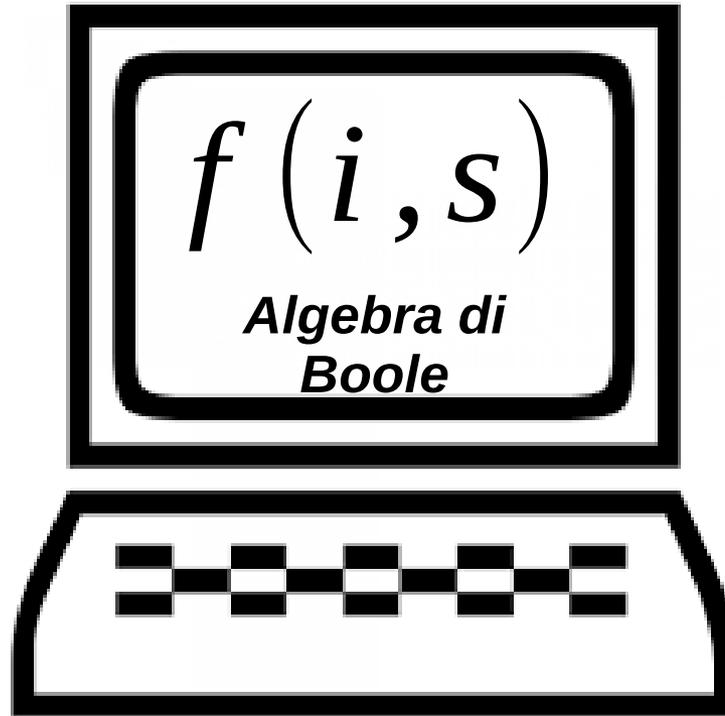
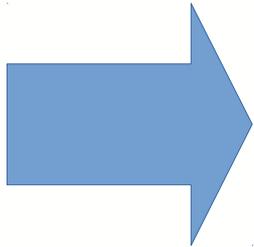
- Dati
- Controlli

Uscite Principali:

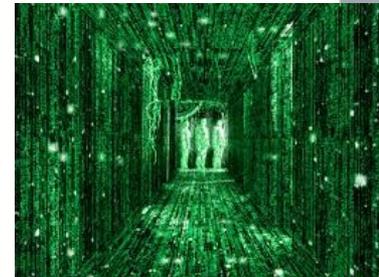
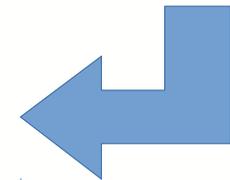
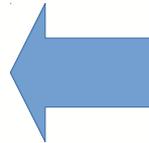
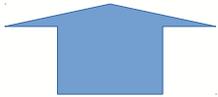
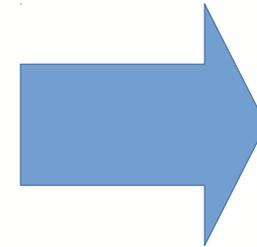
- Dato
- Stato

L'elaboratore

I: input



O: output



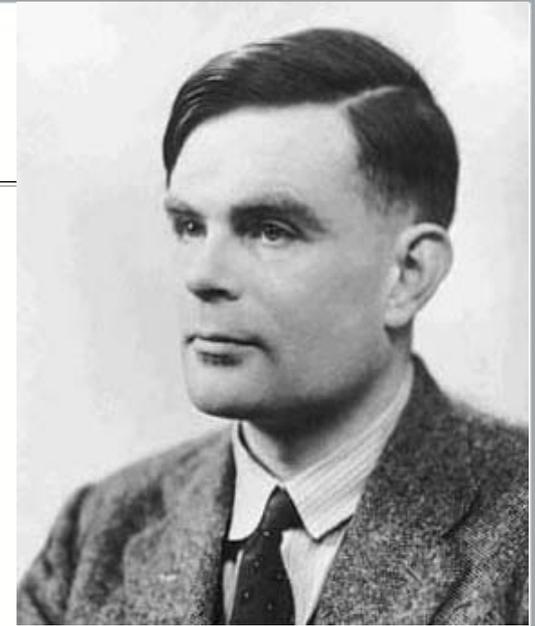
I requisiti

Per realizzare l'esecuzione automatica dell'informazione occorre:

- ***Disporre di un elaboratore***
- Rappresentare l'informazione (dati)
- Rappresentare l'elaborazione (programma)

Il Primo Informatico

- Alan Turing
 - Un matematico Inglese
 - Non un ingegnere
- In breve
 - “On computable numbers, with an application to the Entscheidungsproblem”
 - Enigma (Seconda Guerra Mondiale)
 - Progettazione del primo calcolatore e linguaggi
 - Intelligenza artificiale



**ALAN MATHISON
TURING**

(1912-1954)

Entscheidungsproblem

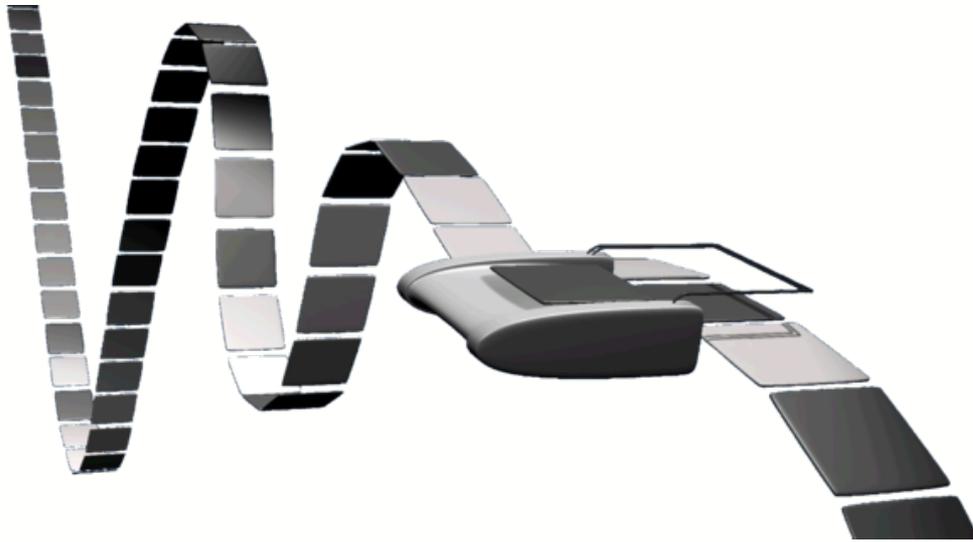
“Trovare una procedura algoritmica per decidere se una qualunque formula nella logica dei predicati è valida”

(p.es. se una qualunque formula dell'aritmetica è un teorema, cioè derivabile dagli assiomi mediante le regole di inferenza)

In un articolo intitolato *On Computable Numbers, with an Application to the Entscheidungsproblem*”, un giovane matematico di nome Turing dimostrò la non esistenza di tale algoritmo.

La macchina di Turing

Alan Turing propose nel 1936 l'idea di una “macchina immaginaria” che potesse effettuare ogni tipo di calcolo su numeri e simboli



COMPLESSITÀ

Cosa si intende per algoritmo complesso

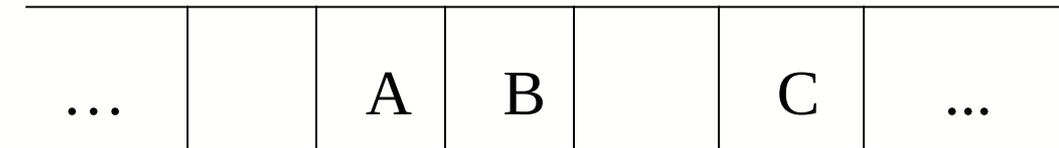
CALCOLABILITÀ

Cosa può fare una macchina

Com'è fatta una MdT?

Una MdT è composta da:

- Un nastro di lunghezza infinita diviso in celle. Ciascuna cella contiene un simbolo di un ben determinato alfabeto finito oppure è vuota



- Una testina che si sposta da una casella all'altra del nastro effettuando operazioni di lettura e scrittura.

Evolutioni (1)

La macchina evolve nel tempo e ad ogni istante si può trovare in uno stato interno ben determinato facente parte di un insieme finito di stati.

Ogni passo dell'evoluzione viene determinato dallo stato attuale s nel quale la macchina si trova e dal carattere c che la testina di I/O trova sulla casella del nastro su cui è posizionata.



Evolutioni (2)

Una evoluzione della macchina consiste in una sequenza di sue possibili configurazioni costituite:

- dallo stato interno attuale

- dal contenuto del nastro (una stringa di lunghezza finita)

- dalla posizione sul nastro della testina di I/O.

Nei casi più semplici l'evoluzione ad un certo punto si arresta in quanto non si trova nessuna istruzione in grado di farla proseguire.

Conclusione elaborazione

Si può avere un arresto in una configurazione "utile" dal punto di vista del problema che si vuole risolvere; in tal caso quello che si trova registrato sul nastro all'atto dell'arresto rappresenta il risultato dell'elaborazione.

Si può avere però anche un arresto "inutile" che va considerato come una conclusione erronea dell'elaborazione.

Può anche accadere che un'evoluzione non abbia mai fine; in pratica la macchina non ha realizzato l'algoritmo.

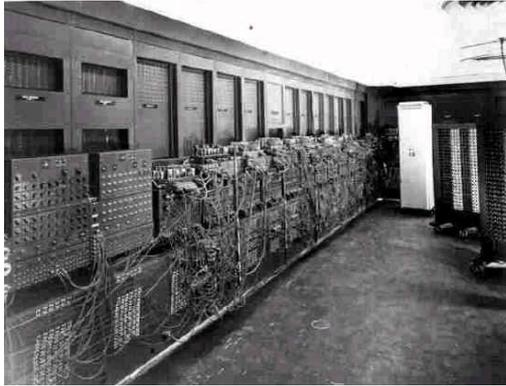
Tesi di Church-Turing (1)

“Data la funzione $Y = f(X)$, esiste sempre una MdT che la calcoli?”

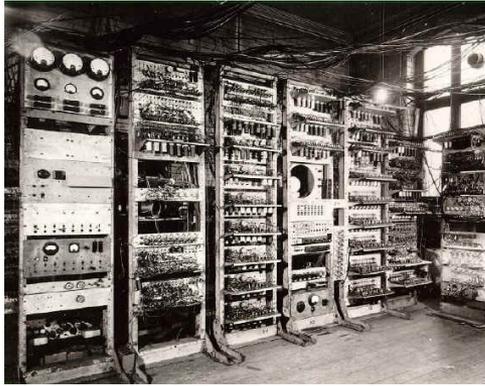
Esistono funzioni non calcolabili dalla MdT

“Se una funzione è non calcolabile secondo Turing, esiste un altro formalismo che la può calcolare?”

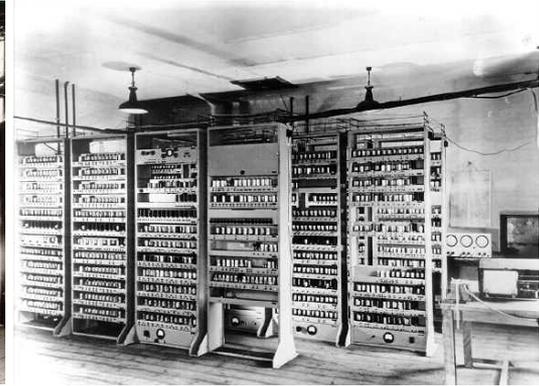
Non esiste un formalismo né una macchina concreta che possa calcolare una funzione non calcolabile secondo Turing



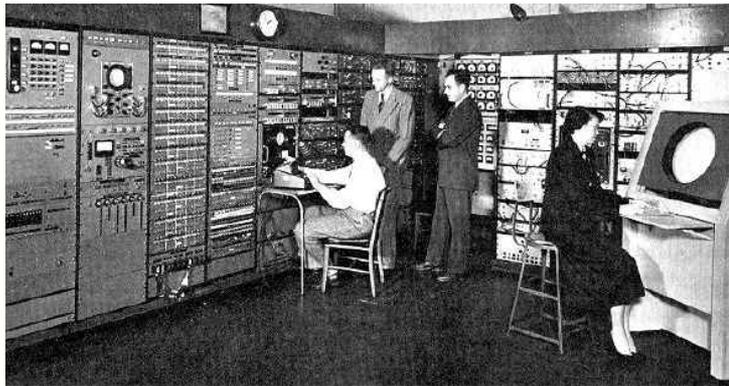
ENIAC (1946)



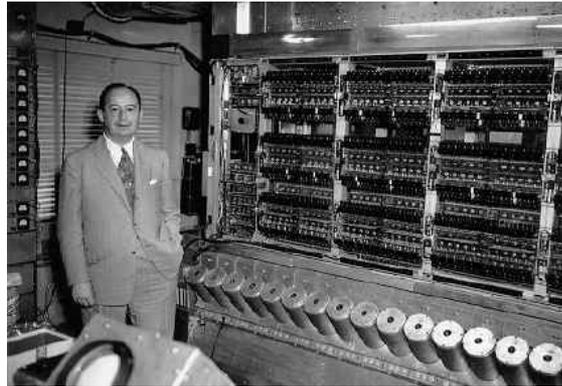
Mark I (1948)



EDSAC (1949)



**Whirlwind
(1949)**



IAS (1952)

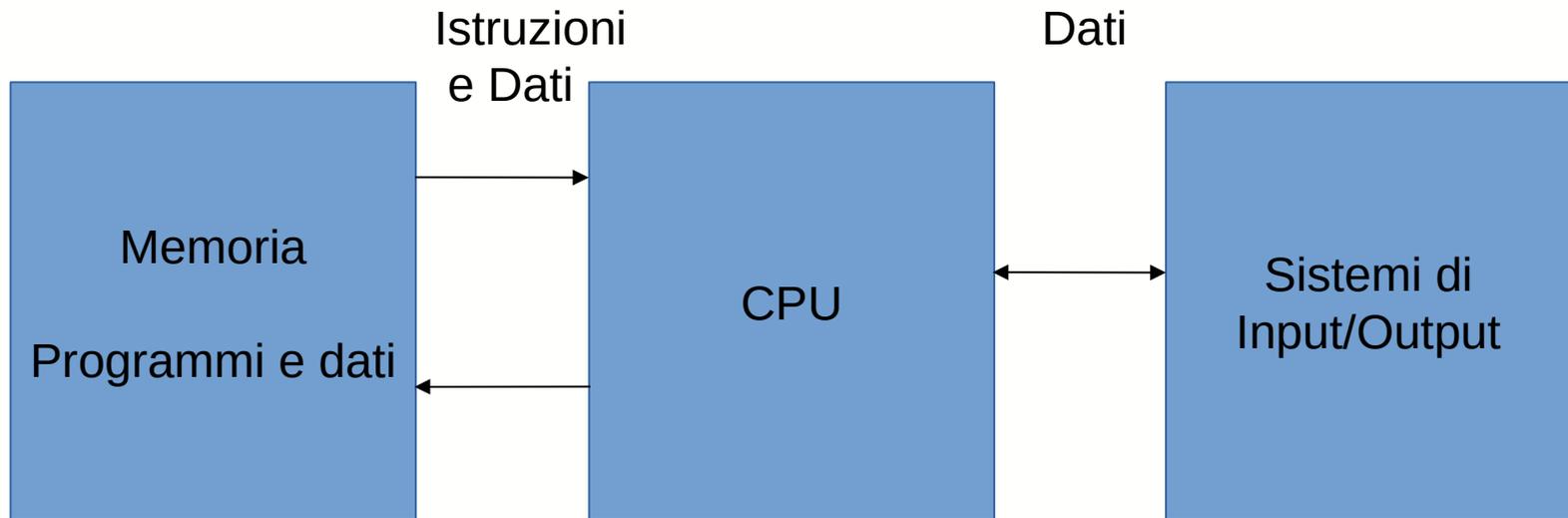


UNIVAC (1952)

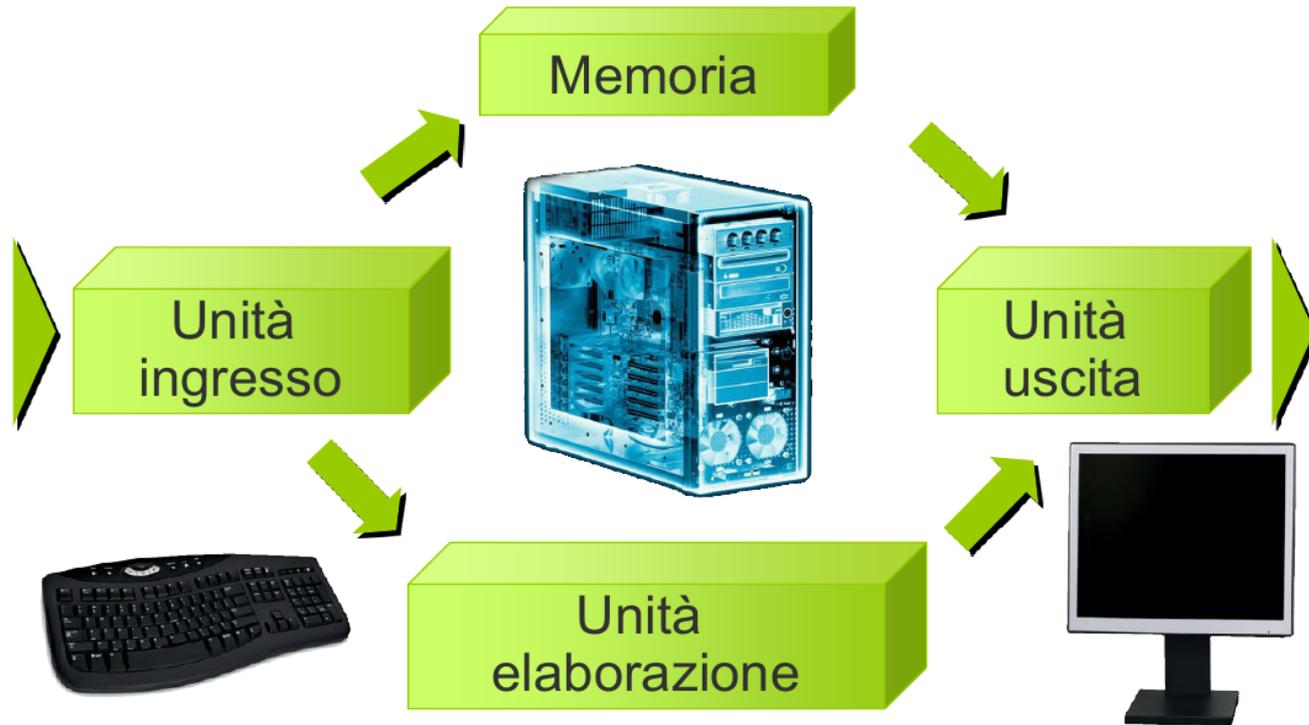
Il Modello di Elaboratore

- Von Neumann
 - Americano
 - Il Modello di Von Neumann
 - Il ciclo di Von Neumann

Modello Von Neumann

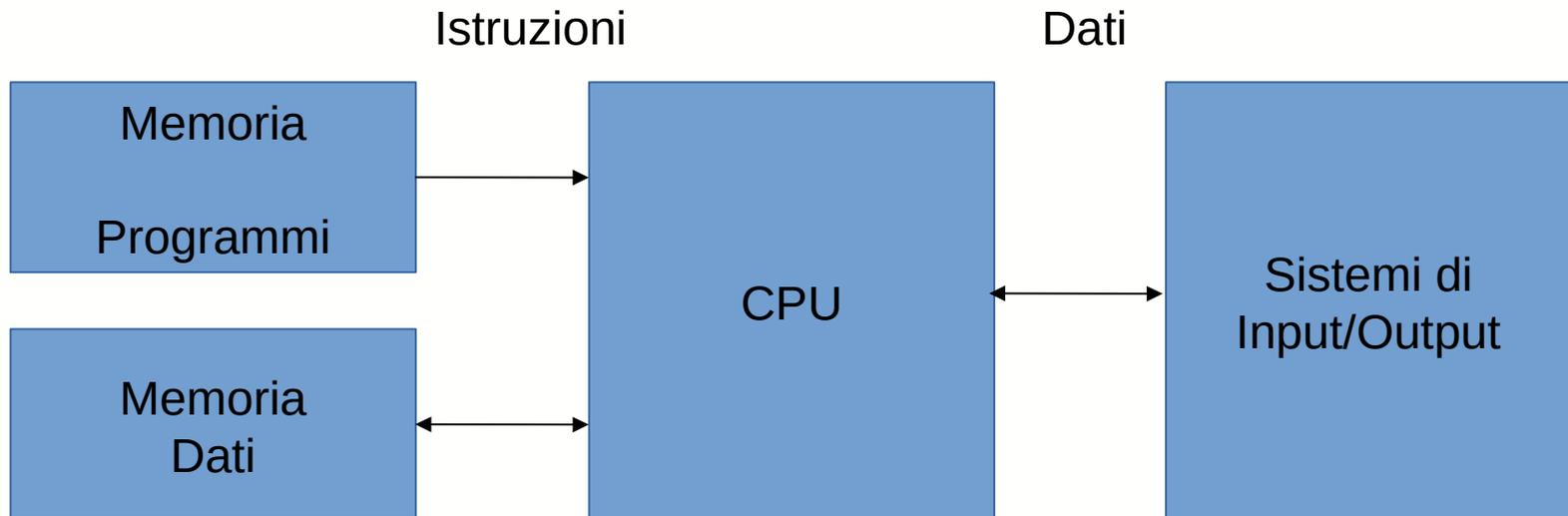


Il Modello di Von Neumann



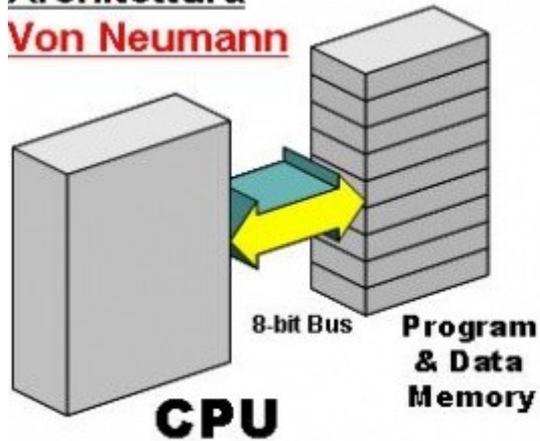


Modello Harvard

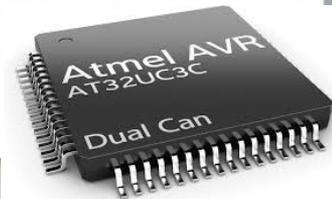
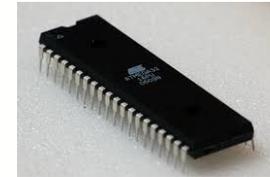
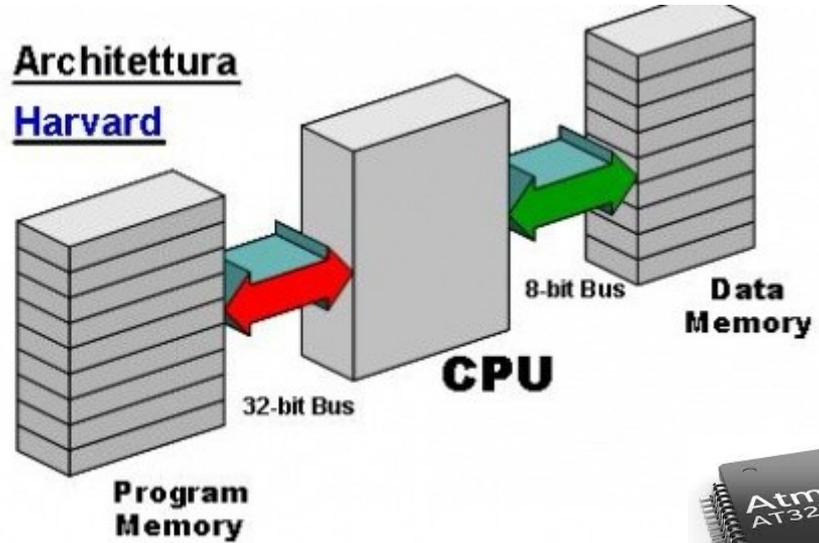


Von Neumann vs Harvard

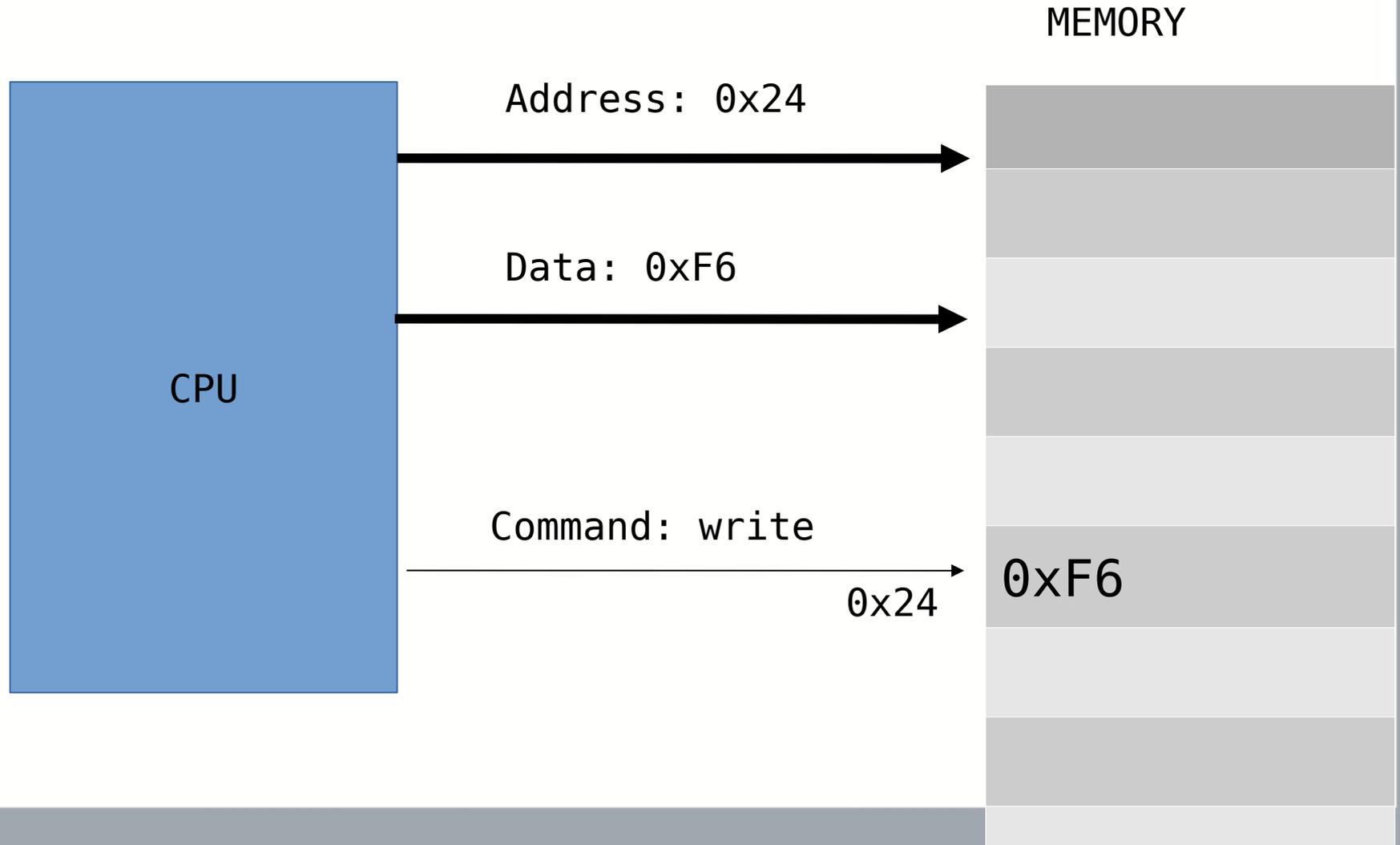
**Architettura
Von Neumann**



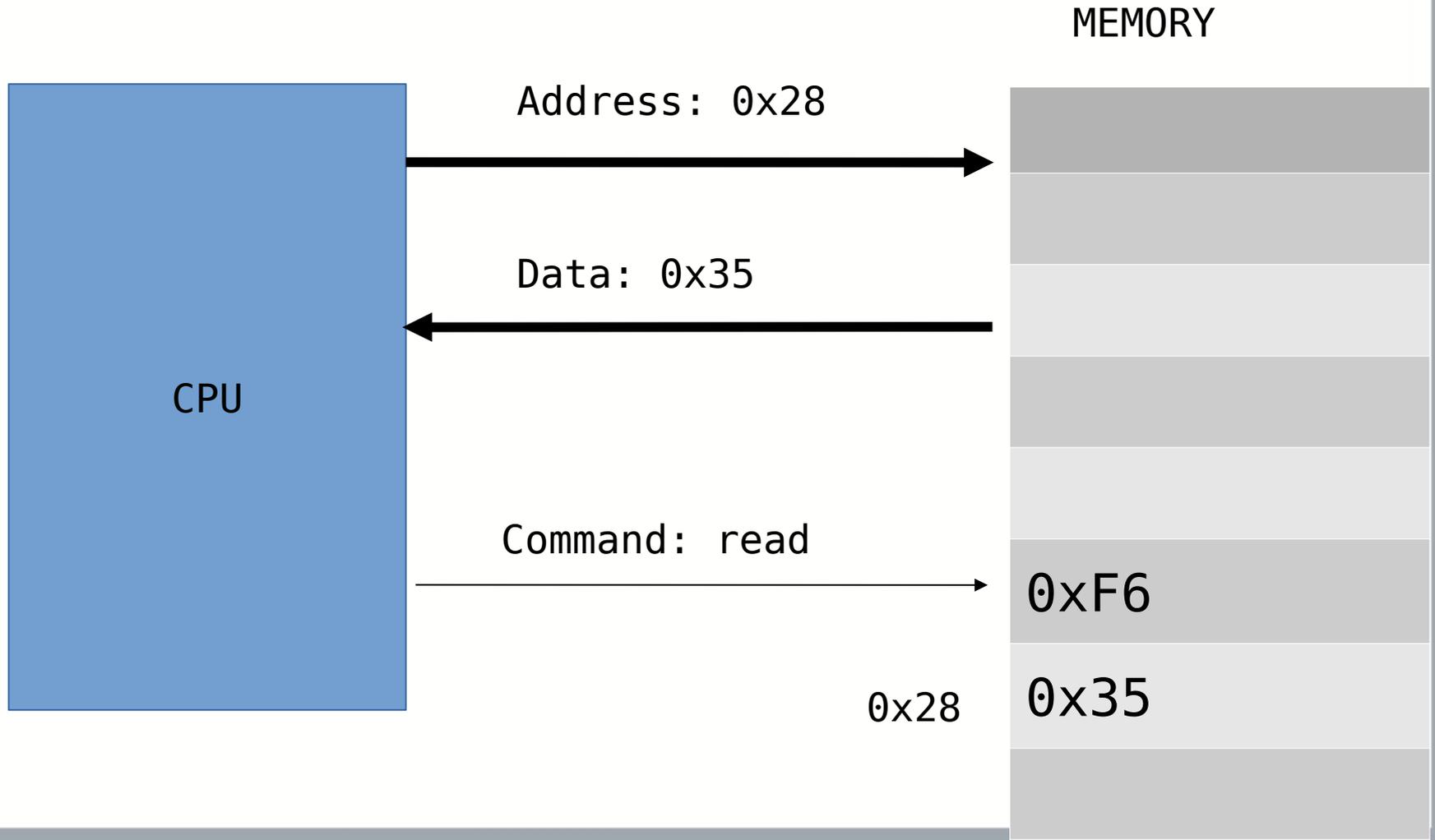
**Architettura
Harvard**



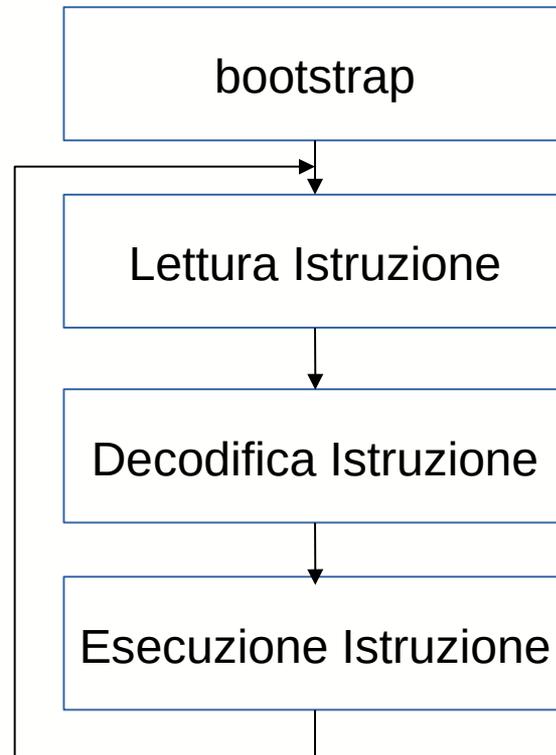
Memory Write



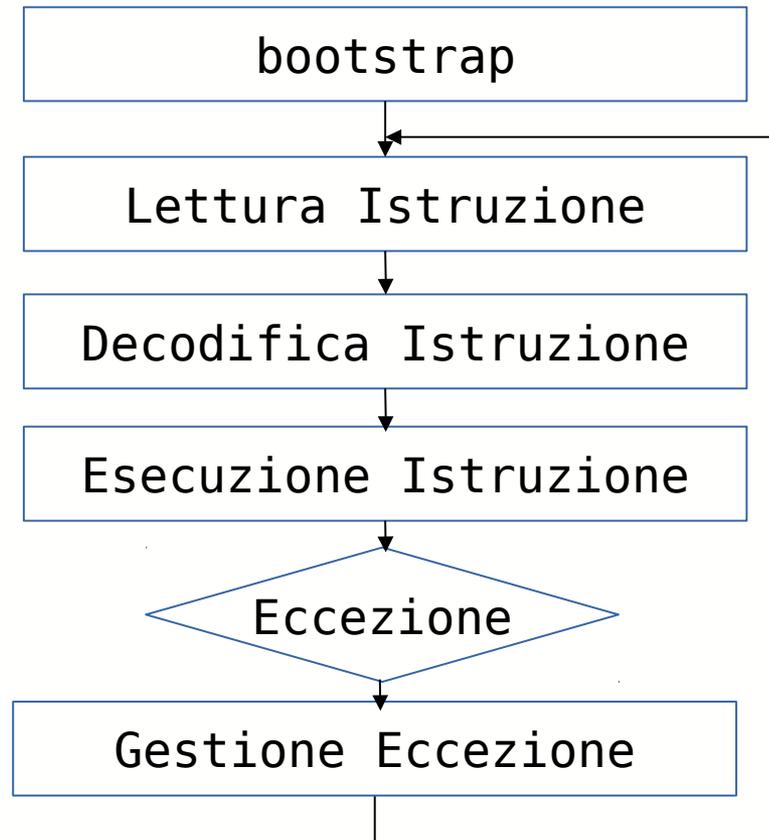
Memory Read



Ciclo di Von Neumann



Gestione Eccezioni



Livelli di un elaboratore (Tanebaum)

