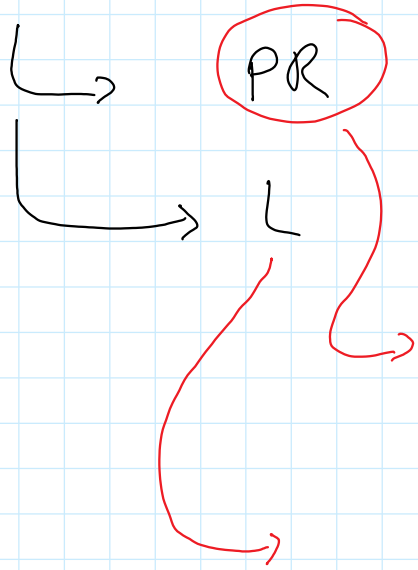


PR L A

lunedì 17 settembre 2018 08:22



9 CFU
3 CFU

25h → 8h aula
17h individ.

ROBERTO BARBUTI
roberto.barbuti@unipi.it
DAVIDE BACCIU

45 minuti
quarto d'ora accademico (pausa
di 15 minuti inizio dell'inizio)

9.00 → 9.15

9.00 - 11.00 →

9.15 - 10.00

10.15 - 11.00

LUN

9.00 - 11.00

MER

11.00 - 13.00

GIO

16.00 - 18.00

16.15 → 17.45

Ricerimenti

GIO

14.00 - 16.00

?

CORSO A

Scritto (1,5 h)

Prova pratica (Lab)

~~Orale (teoria)~~

DOCENTI INFORMATICA

DIP. di Informatica

Edificio C

2° piano

Teoria e uso dei linguaggi di programmazione

Teoria dei linguaggi formali

automi a stati finiti

grammatiche a strutture di frase

Semantica formale (quell'è il significato dei programmi, ovvero le modifiche operate all'interno di un calcolatore durante l'esecuzione dei programmi)

PARADIGMI DI PROGRAMMAZIONE

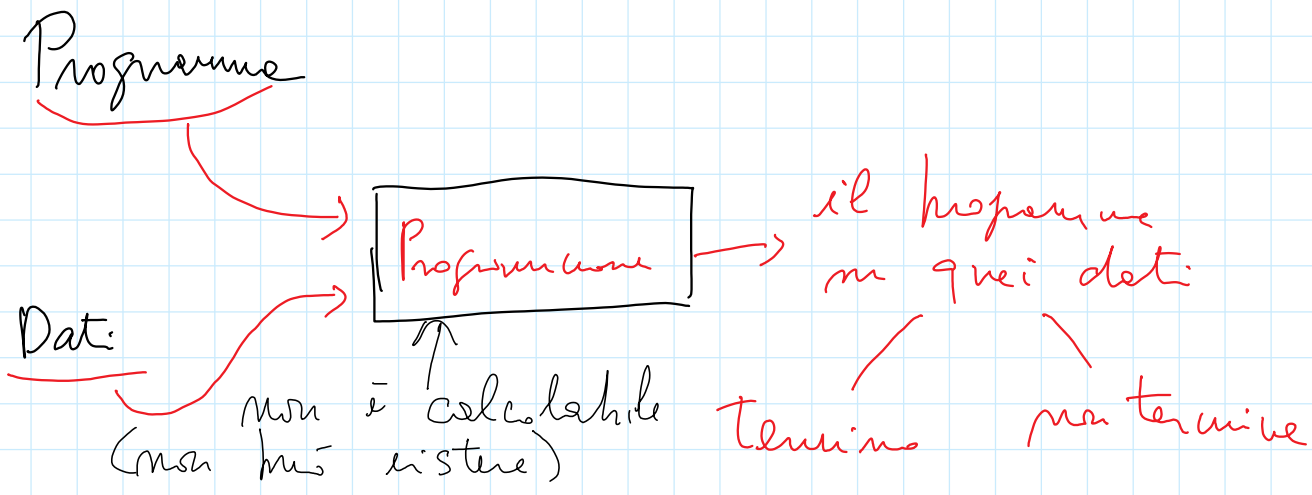
(modi diversi di risolvere un problema)

- paradigma ITERATIVO
 - paradigma RICORSIVO
- } hanno le
stesse
potenze

(non tutti i problemi si possono risolvere con l'informatica)

} Teoria della
Calcolabilità

I programmi hanno "andare in ciclo"
(cioè avere calcoli INFINITI)

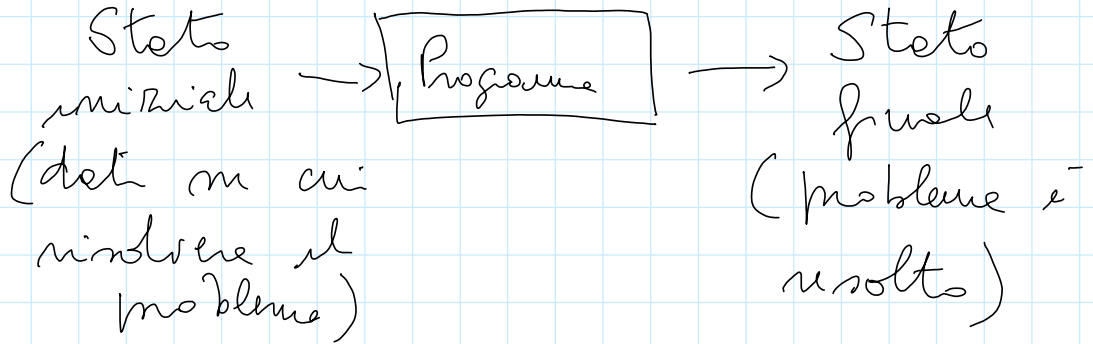


Paradigme

iterativo

i problemi vengono risolti.

attraverso "TRASFORMAZIONI DI STATO"



Come è fatto lo stato?

Come si crea lo stato?

Come si modifica lo stato?

Il linguaggio C è un **LINGUAGGIO IMPERATIVO** (cioè basato sul paradigma iterativo)

PARADIGMA RICORSIVO

lunedì 17 settembre 2018 10:01

Non ha lo stato e i problemi
vengono risolti attraverso la
definizione e chiamata di
FUNZIONI RICORSIVE.

Una funzione ricorsiva è
definita in termini di se stessa.

Linguaggio C (qualcuni manuali)

Il corso è coperto da DISPENSE

(Dati - Informatica → Persone → R. Barbut)

$$! : \mathbb{N} \rightarrow \mathbb{N}$$

$$0! = 1$$

$\uparrow \mathbb{N}$ $\uparrow \mathbb{N}$

$0! = 1$ chiamate (applicazioni)
 $3! = 3 \cdot 2 \cdot 1 = 6$
 $4! = 4 \cdot 3 \cdot 2 \cdot 1 = 24$

$$n! = n \cdot (n-1) \cdot \dots \cdot 1 \quad n \in \mathbb{N}^+$$

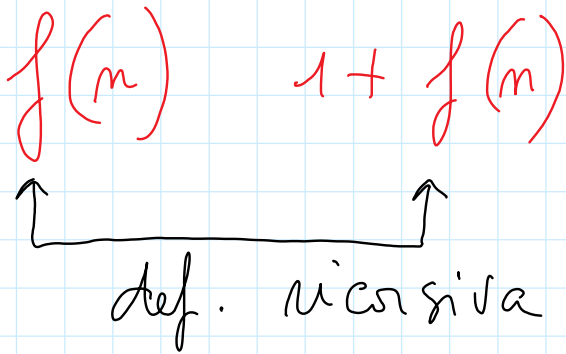
$\mathbb{N} \setminus \{0\}$

definizione

metuali: maggiori di 0

$$f: \mathbb{N} \rightarrow \mathbb{N}$$


tipo delle funzione
(dominio \rightarrow codominio)



$$\left. \begin{array}{l} f(3) \\ = \{ \text{def } f \} \\ 1 + f(3) \\ = \{ \text{def } f \} \\ 1 + 1 + f(3) \\ \vdots \end{array} \right\}$$

Come è fatto lo stato?

Il linguaggio C è "ad alto livello",
cioè i costrutti del linguaggio sono
"comprensibili" per il programmatore.

Il calcolatore  *Completion (programmazione)* non è in grado di
eseguire programmi (ma istruzioni
di un "linguaggio macchina")

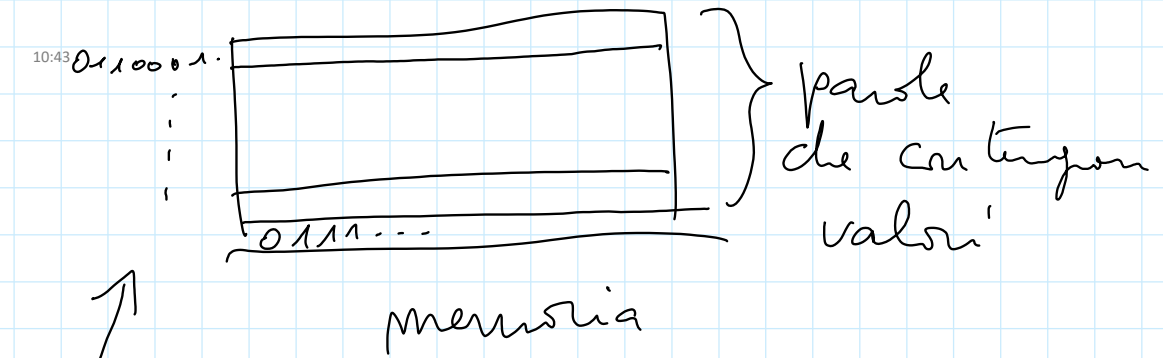
linguaggio macchina ha

istruzioni che sono sequenze di
bit $\{0, 1\}$



questo è l'insieme dei valori
0 e 1

Per spiegare il significato dei
programmi C devo "ASTRARE"
del computer reale e ragionare
in questo LIVELLO DI ASTRAZIONE



indirizzi

che valore contiene
 l'indirizzo 000110?
 Il valore contenuto in
 000110 è 1110010

Metti nelle parole di memoria
 all'indirizzo 000110 il
 valore 000000.

Qual'è il valore all'indirizzo
 000110?
 000000

Memoria estetica

lunedì 17 settembre 2018 10:47

x	15 30
y	20
z	30
w	4

Che valore ha x? 15

Modifica il valore di x con 30!

Lo stato nel quale diamo significato ai programmi C è uno stato astratto composto da associazioni nome - valore

x	30
y	20
z	15
w	4

$\{ \langle x, 30 \rangle, \langle y, 20 \rangle, \langle z, 15 \rangle, \langle w, 4 \rangle \}$

Lo stato rispetto al quale voglio
dare significato ai programmi C
non è la memoria fisica del calcolatore,
ma un insieme di associazioni
nome - valore.

TROPPO SEMPLICE

COMPLI CHIAMO "NOW POINT"

lunedì 17 settembre 2018 10:55

AMBIENTE

x	l ₀
y	l ₁
z	l ₂

MEMORIA

l ₀	30
l ₁	20
l ₂	15

nomi indirizzi
di memoria
(astratti)
simbolici

l ste ha locazione (indirizzamento di memoria)

Il valore del nome x lo trovo in memoria
all'indirizzo associato a x nell'ambiente