

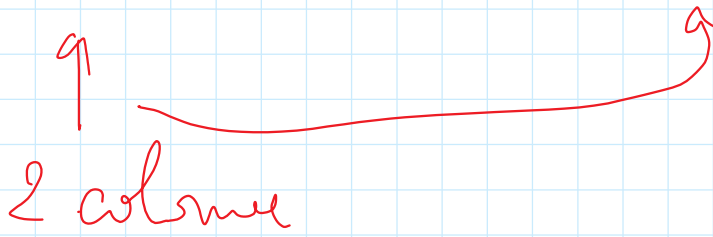
ambiente

x	l ₀
y	l ₁

memoria

l ₀	15
l ₁	10

← tante
righe
(tante
quante
sono le
variabili)



COMANDI - istruzioni del ling. di prog.
che modificano lo stato (modificano
soltamente la memoria)

ASSEGNAZIONE - modifica il valore di una variabile

x	l0
y	l1

l0	15
l1	10

$x = x + 2;$
↑
la variabile da modificare

←
l'espressione che dà il valore con il quale modificare la variabile

x	l0
y	l1

l0	17
l1	10

x	15
y	10

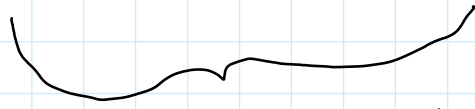
stato

$x = x + 2;$

x	17
y	10

troppo astratto!

$\{c_1; c_2; \dots; c_m;\}$



una sequenza di comandi
diventa un unico comando

x	10
y	11

l0	15
l1	10

$\left\{ \begin{array}{l} x = x + 1; \\ y = y + x; \end{array} \right.$

x	11
y	22

l0	16
l1	26

if condizione then comando1 e comando2
↑
espressione
logica
(cioè che
vale vero
o falso)
↑
generi
comandi

Se la condizione è vera viene eseguito il comando1, se è falsa comando2

In C:

if (condizione) comando
se la condizione è vera viene eseguito "comando", altrimenti il comando condizionale lascia lo stato invariato.

if (condizione) comando1 e comando2
↑
vera
↑
falsa

falsa

COMANDO ITERATIVO

while (condizione) comando

Viene valutata la condizione:

- Se è falsa il comando while termina
- Se è vera viene eseguito "comando" e poi viene eseguito di nuovo il comando while nello stato raggiunto

x	l ₀
y	l ₁

l ₀	3
l ₁	∅

```
while (x > 0) { x = x - 1; y = y + 2; }
```

Corpo del while

guardia

--

l ₀	2
l ₁	2

--

l ₀	1
l ₁	4

x	l ₀
y	l ₁

l ₀	∅
l ₁	6

x	l₀
y	l ₁

l₀	3
l ₁	0

while (x > 0) { x = x + 1;
y = y + 2; }

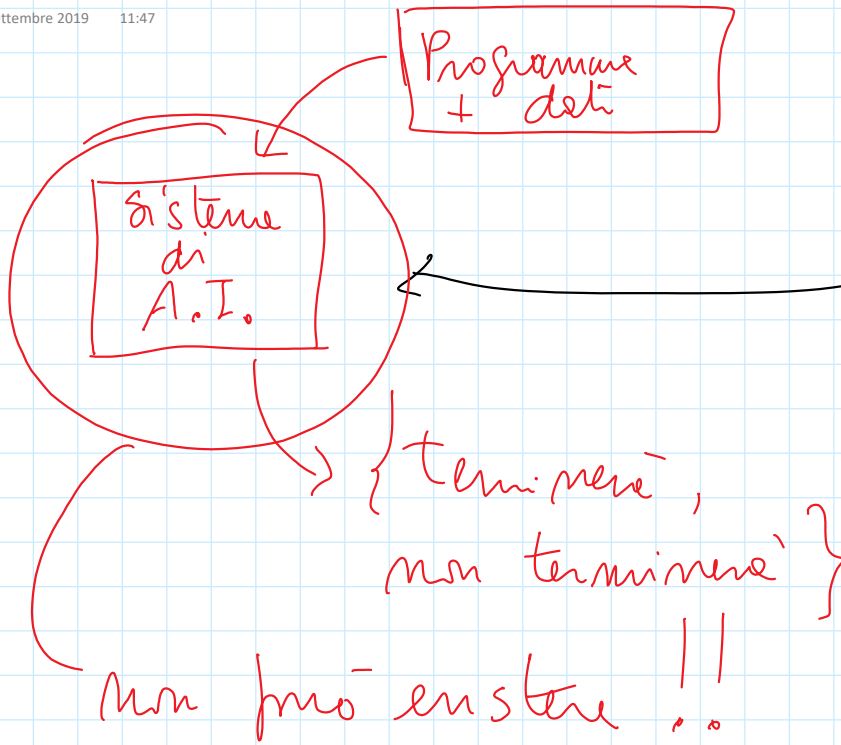
//

l ₀	4
l ₁	2

0
0
0
0

NON TERMINA

"andata in ciclo"



teoria della
Calcolabilità,
Il problema
della TERMINAZIONE
DEI PROGRAMMI
NON È
CALCOLABILE

TOGLIAMO DAL LINGUAGGIO il
comando while.

Aggiungiamo un comando `stretto`
che termina sempre! (for)

NON SI PUO' FARE: se eliminiamo
la possibilità di avere calcoli
infiniti, non possiamo risolvere
TUTTI i problemi risolvibili.
Con l'informatica.

lo stato di un programma si
crea attraverso dei costrutti che
ni chiamano

DICHIARAZIONI

tipo nome ;
tipo nome = espressione ;

int x ;
int y = 10 ;

3.2

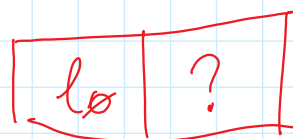
↳ tipo è il tipo di valori
che le variabili potranno
assumere

int vuol dire valori interi

ambiente

memoria

int x ;



le
dichiarazioni
creano
0 - < t - t

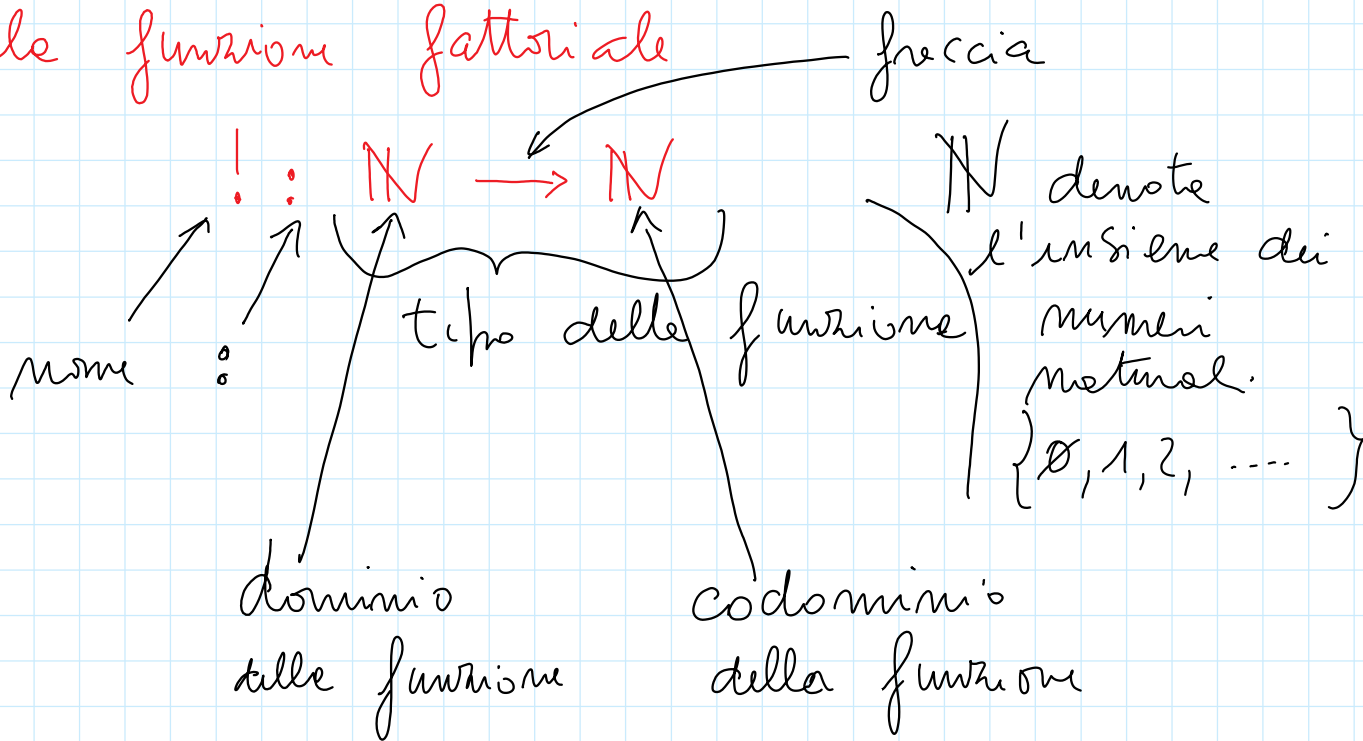
`int y = 10;`

x	l ₀
y	l ₁

l ₀	?
l ₁	10



le funzioni fattoriali



$$\begin{cases} 0! = 1 \\ n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 1 \quad \text{se } n > 0 \end{cases}$$

$$3! = 3 \cdot 2 \cdot 1 = 6$$

$$5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$$

$$0! = 1 ;$$

fattoriale del valore delle variabile n

```
int n = 10;  
int m = 5;
```

No

```
int n = 10;  
m = 5;
```

Contatto!

```
int n = 5;  
int f = 1;  
while (n > 0)
```

```
{ f = f * n;  
  n = n - 1;  
}
```

moltiplicazione

il risultato è il valore delle variabile f.

int m = 4;

m	l ₀	l ₀	4
---	----------------	----------------	---

int f = 1;

m	l ₀	l ₀	4
f	l ₁	l ₁	1

m > 1
m >= 2

while (m > 0) {
 f = f * m;
 m = m - 1;
}

--

l ₀	3
l ₁	4

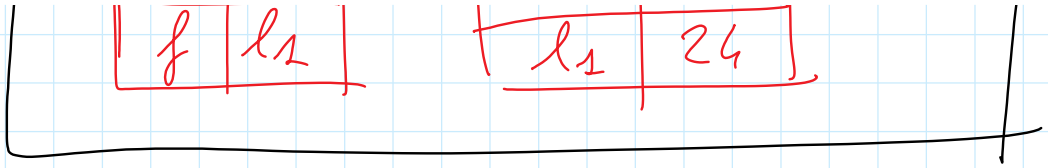
--

l ₀	2
l ₁	12

--

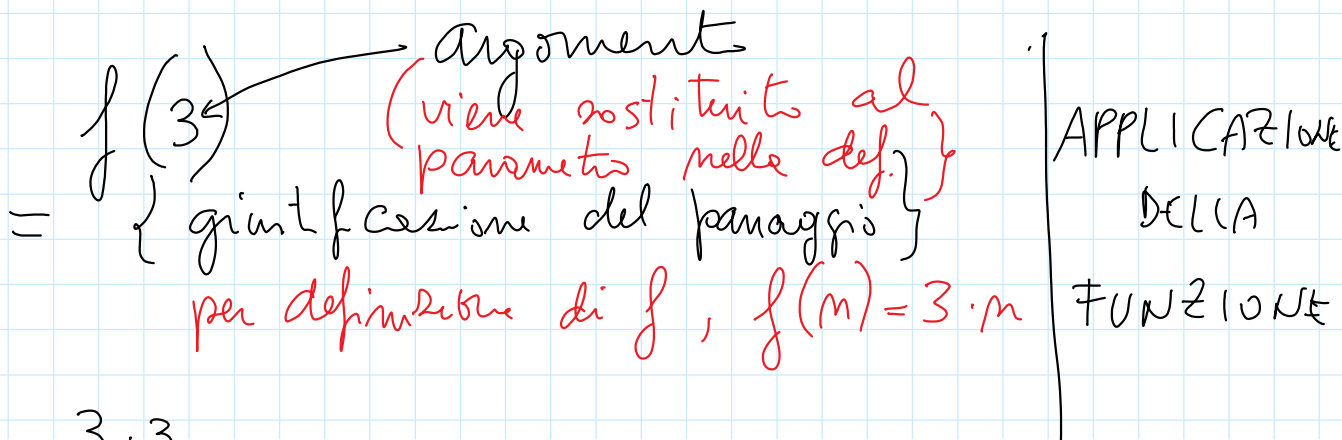
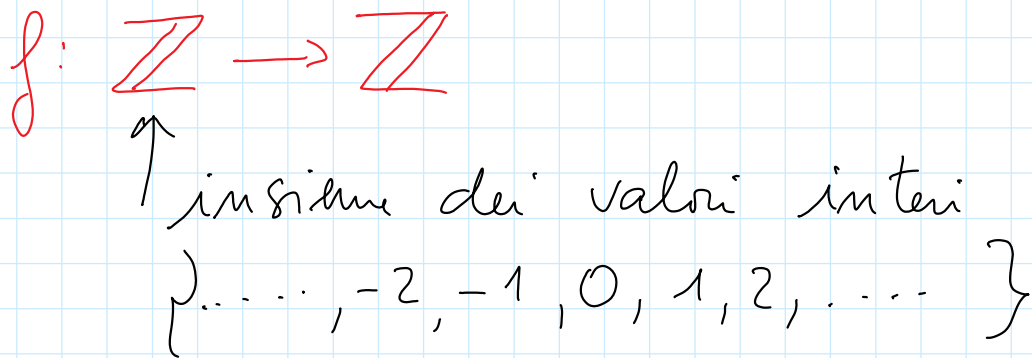
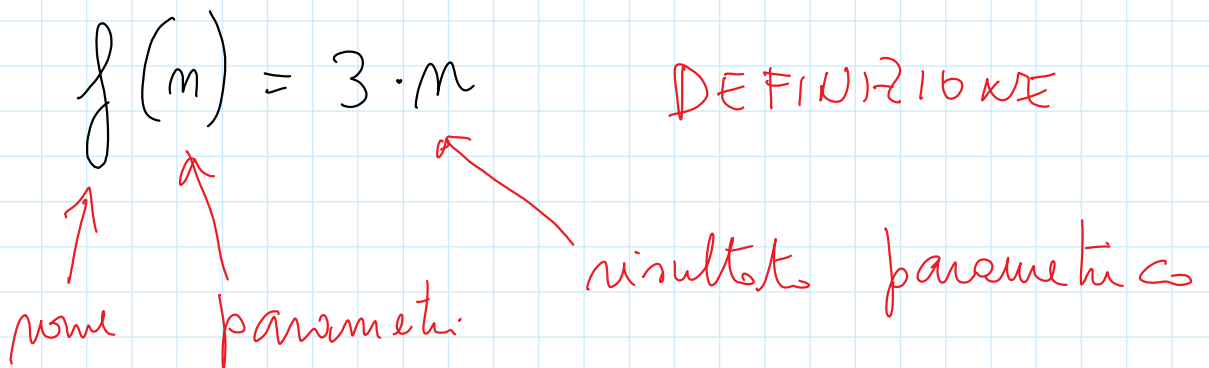
l ₀	1
l ₁	24

m	l ₀	l ₀	0
f	l ₁	l ₁	24



PARADIGMA RICORSIVO (linguaggio funzionale)

programmi sono
definizioni di funzioni } ?
applicazioni di funzioni }



$$= \frac{3.3}{9} \left\{ \text{calcolo} \right\}$$

0 0

|

ML

↳ Meta Language
veniva usata, data la sua
semplicità, per descrivere
la semantica di altri
linguaggi.

→ F# Microsoft

→ CAML
(Camel)

in CAML si definiscono e si
applicano funzioni.

let $f(m) = 3 * m$;;

$f: \text{int} \rightarrow \text{int}$

← usate CAML

inferenza dei tipi

$f(n) \dots$

definizione

$f(3) ::=$

$- : \text{int} = 9$

definizione

applicazione

$$\text{let } f(n) = n * (n-1) * \dots * 1 ; ;$$

NO

Possibilità di definire
funzioni ricorsive

Sono funzioni che nelle loro
definizioni applicano le funzioni
che stanno definendo.

definizione ricorsiva della funzione fattoriale.

$$0! = 1$$

$$n! = n * (n-1)! \quad \text{se } n > 0$$

↑
 utilizzo ! nella mia definizione

$$4! = \{ \text{def } !, 4 > 0 \}$$

$$4 * 3!$$

$$= \{ \text{def } !, 3 > 0 \}$$

$$4 * 3 * 2!$$

$$= \{ \text{"}, 2 > 0 \}$$

$$4 * 3 * 2 * 1!$$

$$= \{ \text{"}, 1 > 0 \}$$

$$4 * 3 * 2 * 1 * 0!$$

$$= \{ \text{def } !, 0 \neq 0 \}$$

$$4 * 3 * 2 * 1 * 1$$

let rec f(n) =

if n > 0 then n * f(n-1)
 else 1 ;;

f: int → int

f(4);;

-: int = 24

= {Calcolo}
24

Problema

MCD (m, m)

$$\text{MCD}: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$$



prodotto cartesiano
operazione tra insiemi

definizione

$$A \times B = \{ (a, b) \mid a \in A \wedge b \in B \}$$

appartenenza ad un insieme

insieme

Coppia di valori

tale che

and

$$A = \{ a, b \}$$

$$B = \{ b, c \}$$

$$A \times B = \{ (a, b), (a, c), (b, b), (b, c) \}$$