

# **Informatica**

## **e Tecnologie della Comunicazione Digitale**

*Docente:*

**Miguel Ceriani ([ceriani@di.uniroma1.it](mailto:ceriani@di.uniroma1.it))**

*Lezioni:*

**Mercoledì/Giovedì/Venerdì 9-11**

*Ricevimento (su appuntamento):*

**Mercoledì 14-16 a viale Regina Elena 295, palazzina F, 1° piano**

# Lezione 3:

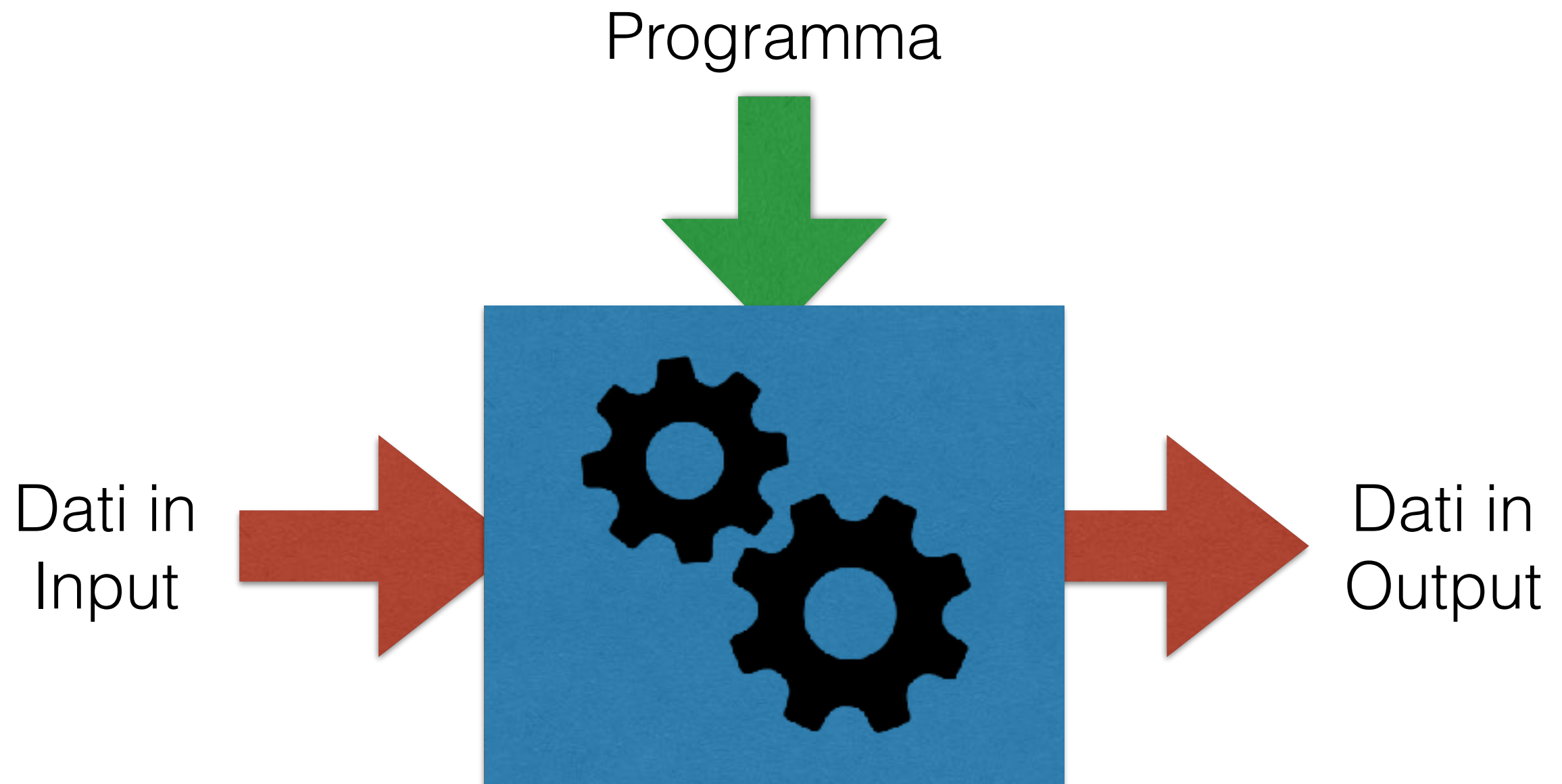
## Bit, Combinazioni, Testo

Nelle puntate  
precedenti...

# Computer:

## Sistema di Elaborazione Dati

### Programmabile Universale



# **Programma:**

## Sequenza Finita di Istruzioni Elementari

1. ripeti finché non hai la mozzarella
  1. trova una charcuterie
  2. fatti dare una mozzarella di Aversa
  3. premi la mozzarella con due dita
  4. cola il latte?
    - se sì, comprane mezzo chilo
2. torna a casa

# bit

quantità più piccola di  
informazione che  
posso considerare  
(2 possibilità, 0 o 1)

# Sistema di Numerazione Binario

x8	x4	x2	x1
1	1	0	1

$$1101_{(2)} = 13_{(10)}$$

# bit

usandone tanti insieme posso rappresentare “qualunque” cosa:

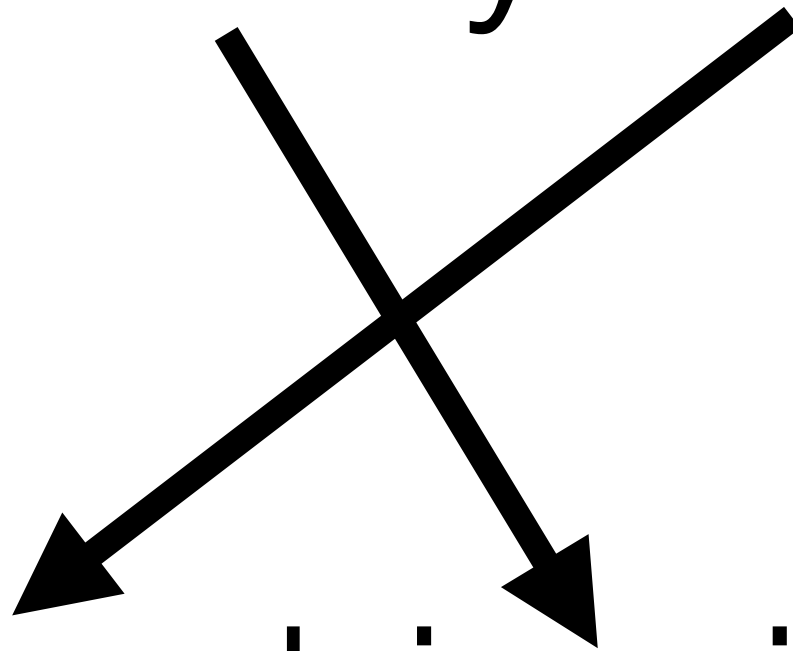
- numeri
- testi
- immagini
- video
- programmi
- ...



bit: **bi**.....**it**

bit: **binary digit**

cifra binaria



# Combinazioni con i Bit

num. di bit	combinazioni
0	1
1	2
2	4
3	8
...	...
$n$	$2^n$
...	...

# Combinazioni di $n$ bit

- dico che per  $n$  bit le combinazioni possibili sono  $2^n$
- ma come dimostrarlo per ogni possibile  $n$  (infiniti)???

# Dimostrazione per Induzione

$n$ : numero di bit

$c_n$ : numero di combinazioni

vogliamo dimostrare che  $c_n = 2^n$ , sempre

- caso base, dimostriamo che  $c_1 = 2^1$ 
  - con 1 bit le possibili combinazioni sono 2, quindi  $c_1 = 2 = 2^1$ .
- induzione, dimostriamo che se  $c_n = 2^n$  allora  $c_{n+1} = 2^{n+1}$ 
  - aggiungendo un bit, che può assumere due valori (0 o 1) il numero di combinazioni raddoppiano, quindi  $c_{n+1} = 2 \times c_n$
  - ma  $c_n = 2^n$ , quindi  $c_{n+1} = 2 \times 2^n = 2^{n+1}$ .

# Gruppi (Multipli) di Bit

byte	8 bit	8 bit
kilobyte (KB, K)	1.000 byte	1.024 ( $2^{10}$ ) byte
megabyte (MB, M)	1.000.000 byte	1.048.576 ( $2^{20}$ ) byte
gigabyte (GB, G)	1.000.000.000 byte	1.073.741.824 ( $2^{30}$ ) byte
terabyte (TB, T)	1.000.000.000.000 byte	1.099.511.627.776 ( $2^{40}$ ) byte

Testo?

# Rappresentazione del Testo

Ciao Mondo!

01000001101101001...





# ASCII Code: Character to Binary

0	0011 0000	O	0100 1111	m	0110 1101
1	0011 0001	P	0101 0000	n	0110 1110
2	0011 0010	Q	0101 0001	o	0110 1111
3	0011 0011	R	0101 0010	p	0111 0000
4	0011 0100	S	0101 0011	q	0111 0001
5	0011 0101	T	0101 0100	r	0111 0010
6	0011 0110	U	0101 0101	s	0111 0011
7	0011 0111	V	0101 0110	t	0111 0100
8	0011 1000	W	0101 0111	u	0111 0101
9	0011 1001	X	0101 1000	v	0111 0110
A	0100 0001	Y	0101 1001	w	0111 0111
B	0100 0010	Z	0101 1010	x	0111 1000
C	0100 0011	a	0110 0001	y	0111 1001
D	0100 0100	b	0110 0010	z	0111 1010
E	0100 0101	c	0110 0011	.	0010 1110
F	0100 0110	d	0110 0100	,	0010 0111
G	0100 0111	e	0110 0101	:	0011 1010
H	0100 1000	f	0110 0110	;	0011 1011
I	0100 1001	g	0110 0111	?	0011 1111
J	0100 1010	h	0110 1000	!	0010 0001
K	0100 1011	I	0110 1001	'	0010 1100
L	0100 1100	j	0110 1010	"	0010 0010
M	0100 1101	k	0110 1011	(	0010 1000
N	0100 1110	l	0110 1100	)	0010 1001
				space	0010 0000

# Quanti caratteri?

- **ASCII** (esteso), è uno **standard USA**, usa **8 bit**, quindi può rappresentare **256 caratteri**: sufficienti per una lingua come Inglese o Italiano ma **pochissimi per tutte le lingue del mondo!!!**
- **UNICODE** è uno **standard internazionale**, usa **da 8 a 24 bit** e mappa al momento **~130.000 caratteri** di lingue di tutto il mondo