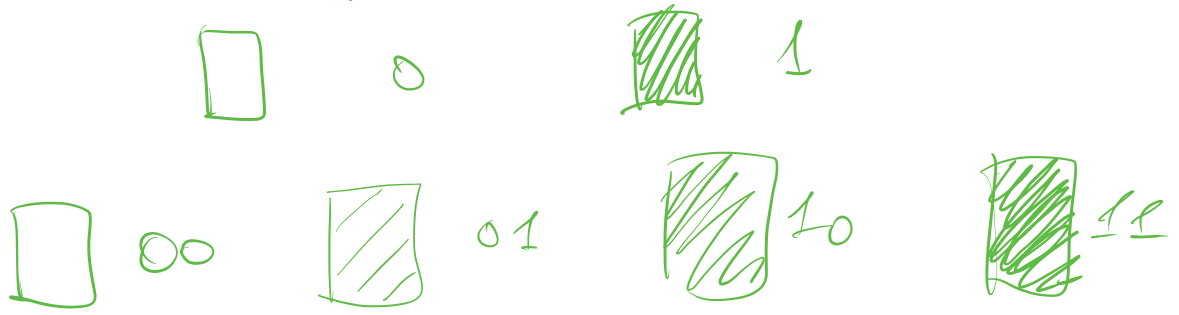
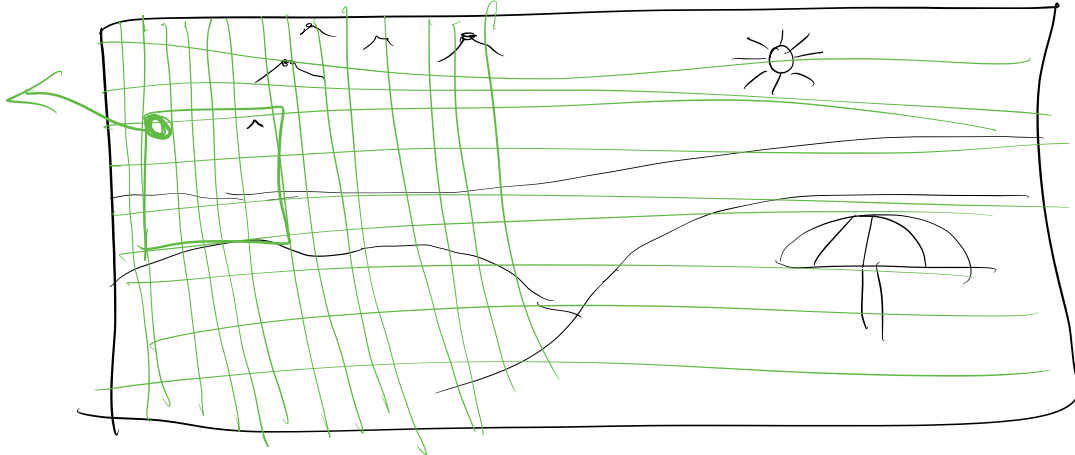


# RAPPRESENTAZIONE DIGITALE DELL'INFORMAZIONE



# TIPICI DI BASE

## TIPICI DI BASE

INTERI

SENZA  
SEGNO

uint, uint8, uint16, uint32,  
uint64

CON  
SEGNO

int, int8, int16, int32,  
int64

FLOATING  
POINT

REALI

float32, float64

COMPLESSI

complex64, complex128

ALTRI

bool

uintptr

byte

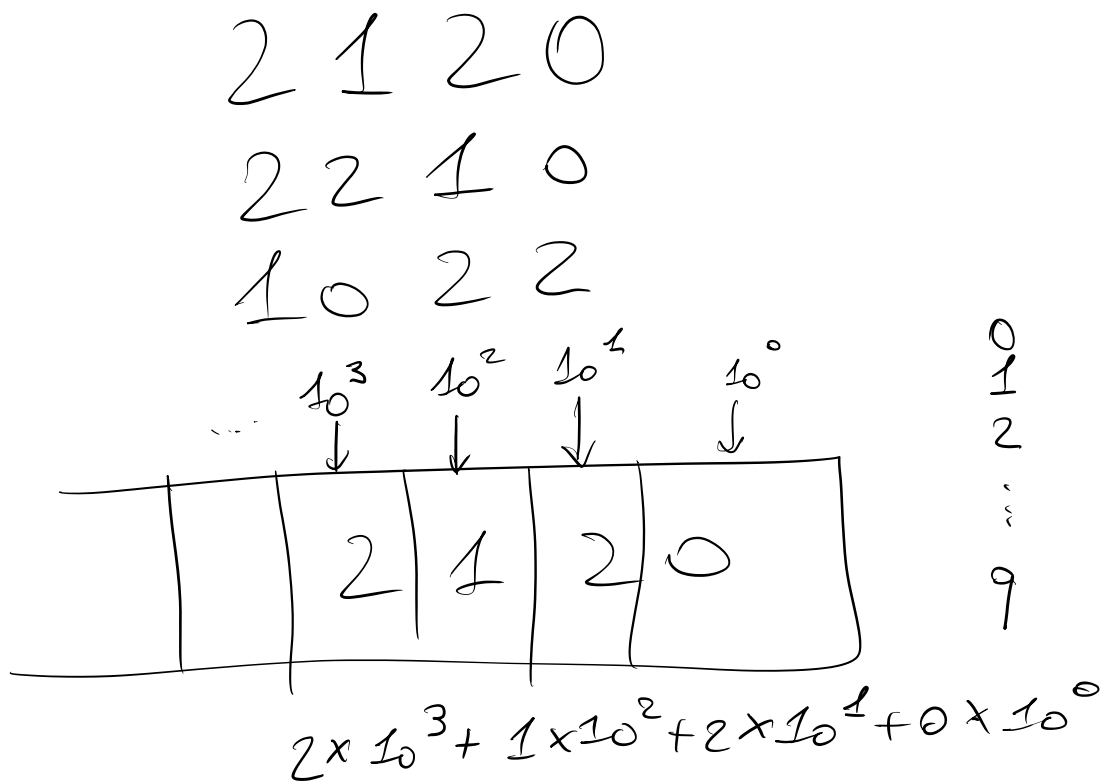
rune

string

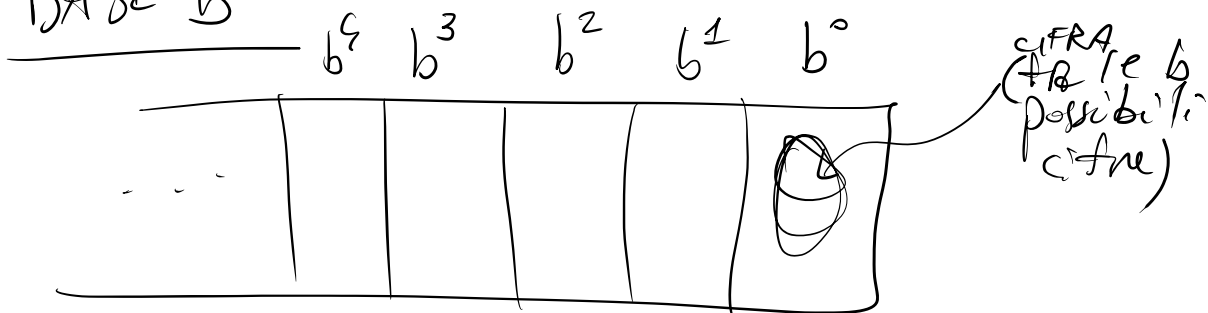
error

INTERI SENZA  
SEGNO

NUMERO vs. NUMERALE  
rappresentazione  
posizionale in  
base 10 → 127  
CXXVII



BASE b

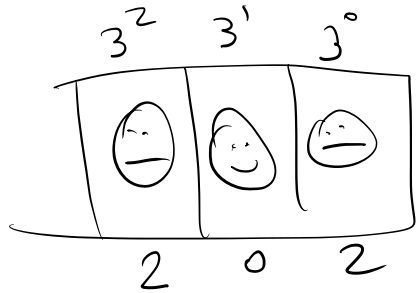


$$b = 3$$

😊 0

☹️ 2

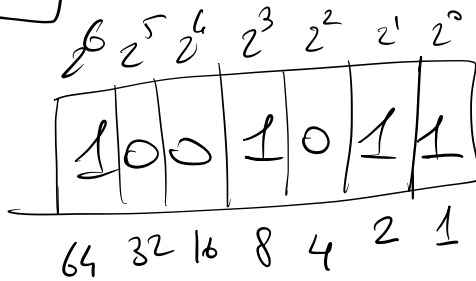
😞 1



$$2 \times 3^2 + 0 \times 3^1 + 2 \times 3^0 = 18 + 2 = 20$$

$$b = 2$$

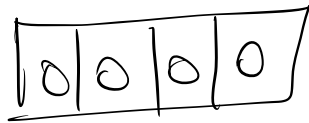
0 1



$$64 + 8 + 2 + 1 = 75$$

128	1
64	1
32	0
16	0
8	0
4	0
2	0
1	1

1	0	0	0	0	1	1
$2^7$					$2^1$	$2^0$
128					4	1
					2	1



0

Overflow

- 0000      0
- 0001      1
- 0010      2
- 0011      3

⋮

1111

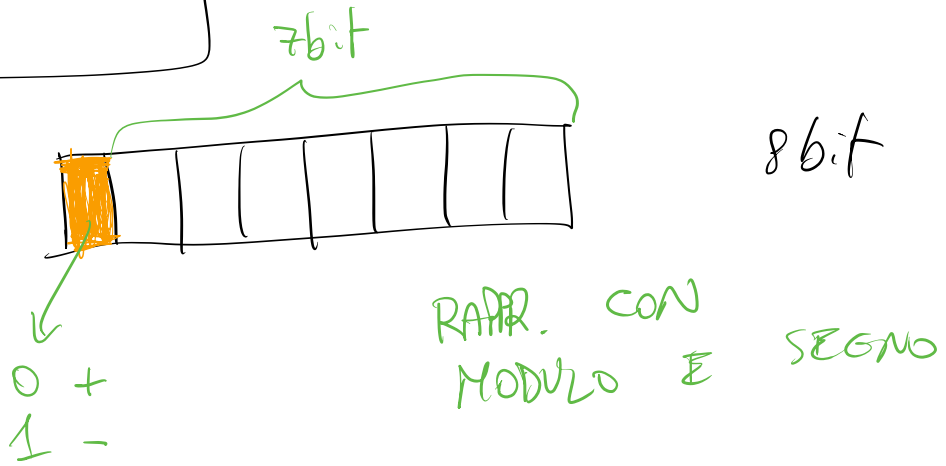
$$15 = 2^4 - 1$$

$2^3 \ 2^2 \ 2^1 \ 2^0$   
8 4 2 1

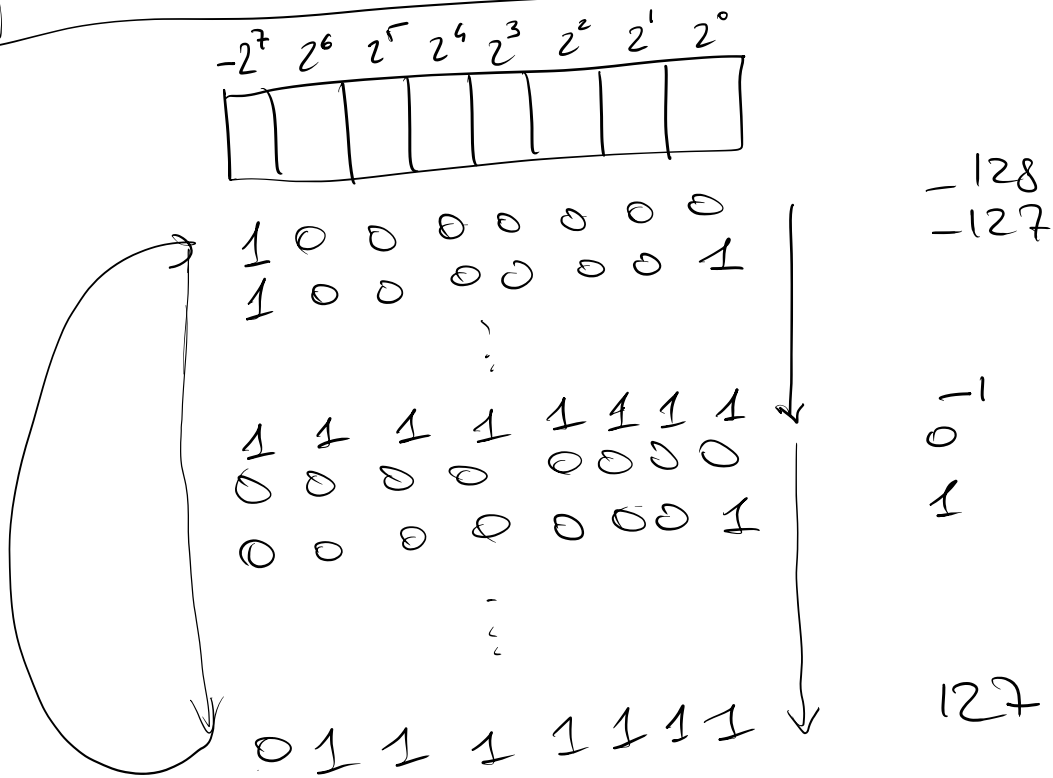
TIPO	N° BIT occupati	N° BYTE	RANGE
<u>uint8</u>	8	1	0 → 255
<u>uint16</u>	16	2	0 → 65.535
<u>uint32</u>	32	4	0 → 4.294.967.295
<u>uint64</u>	64	8	0 → $\approx 1.84 \cdot 10^{19}$
<u>uint</u>	(IMPL. DEP.)		

+      -      \*      /      %

INTERI CON SEGNO



RAPP. IN COMPLEMENTO A 2



TIPO	BIT	BYTE	RANGE
<u>int8</u>	8	1	-128 → 127
<u>int16</u>	16	2	-32768 → 32767
<u>int32</u>	32	4	-2 147 483 648 → 2 147 483 647
<u>int64</u>	64	8	≈ -9 · 10 <sup>18</sup> → 9 · 10 <sup>18</sup>
<u>int</u>	(IMPL. DEP.)		

<u>var</u>	x, y	<u>uint8</u>
<u>var</u>	z, w	<u>uint16</u>
<u>var</u>	i	<u>int</u>

...

$i \leftarrow (z+w) * (x/y)$

$$i = \text{int} \left( (z+w) * \text{uint16}(x/y) \right)$$

$$i = \text{int} \left( \text{uint8}(z+w) * (x/y) \right)$$

$$i = \text{int}(z+w) * \text{int}(x/y)$$



$\underbrace{1/1/1970} \longrightarrow \text{DATA}$   
 EPOCH

1 1 1970  
 25 10 2023

func daysFromEpoch (d, m, a int) int

```

var res int
for i := 1970; i < a; i++ {
  if isleap(i) {
    res += 366
  } else {
    res += 365
  }
}

for i := 1; i < m; i++ {
  if i == 11 || i == 4 || i == 6 || i == 9 {
    res += 30
  } else if i == 2 {
    if isleap(a) {
      res += 29
    } else {
      res += 28
    }
  } else {
    res += 31
  }
}
  
```

```
    }  
    rest = d  
    return res  
}  
func isLeap (a int) bool {  
    return a % 4 == 0 && (a % 100 != 0  
        || a % 400 == 0)
```

```
}
```