Ricapitolando

- I computer sanno eseguire solo un numero molto ridotto di operazioni su simboli di un dato linguaggio (i numeri binari, che vedremo).

- Per funzionare i computer hanno bisogno di:
  - Un programma che li istruisca su cosa fare
  - Dati sui quali agire

Abbiamo quindi bisogno di tradurre sia i programmi che i dati in un linguaggio *"comprensibile"* al computer.

Parleremo di *rappresentazione* e *codifica dell'informazione.*

Gestione delle informazioni

- Nelle attività umane, le informazioni vengono gestite (registrate e scambiate) in forme diverse:
  - idee informali
  - linguaggio naturale (scritto o parlato, formale o colloquiale, in una lingua o in un'altra)
  - disegni, grafici, schemi
  - numeri e codici
- e su vari supporti
  - memoria umana, carta, dispositivi elettronici

- Nelle attività standardizzate dei sistemi informativi complessi, sono state introdotte col tempo forme di organizzazione e codifica delle informazioni
- Ad esempio, nei servizi anagrafici si è iniziato con registrazioni discorsive e poi
  - nome e cognome
  - estremi anagrafici
  - codice fiscale

Informazioni e dati

- Nei sistemi informatici (e non solo), le informazioni vengono rappresentate in modo essenziale, spartano: attraverso i **dati**
- Dal Vocabolario della lingua italiana:

  informazione: notizia, dato o elemento che consente di avere conoscenza più o meno esatta di fatti, situazioni, modi di essere.

  dato: ciò che è immediatamente presente alla conoscenza, prima di ogni elaborazione; (in informatica) elementi di informazione costituiti da simboli che debbono essere elaborati.

Dati e informazioni

- I dati hanno bisogno di essere interpretati.

  Esempio

  `'Mario' '275'` su un foglio di carta sono due **dati**.

  Se il foglio di carta viene fornito in risposta alla domanda "A chi mi devo rivolgere per il problema X; qual è il suo interno?", allora i dati possono essere interpretati per fornire **informazione** e arricchire la **conoscenza**.

- Perché i dati?
  - La rappresentazione precisa di forme più ricche di informazione e conoscenza è difficile.
  - I dati costituiscono spesso una risorsa strategica, perché più stabili nel tempo di altre componenti (processi, tecnologie, ruoli umani).

Codifica dell'informazione

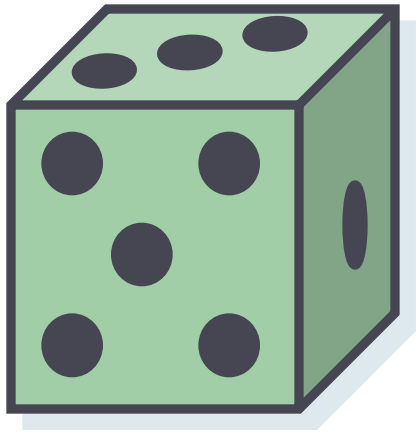
- Per rappresentare l'informazione in modo non fruibile dal computer abbiamo bisogno di codici che la possano esprimere in modo semplice, sintetico e non ambiguo.

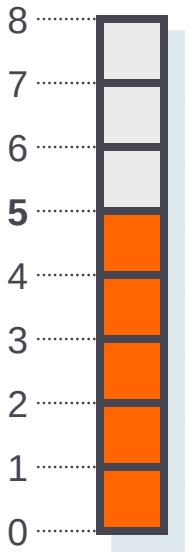
- Un codice (o linguaggio) su basa su:
  - Un alfabeto di simboli
  - Una sintassi con cui comporre tali simboli in sequenze
  - Una semantica da associare ai simboli e alle sequenze

Informazione e supporto

- L'informazione è "**portata da",** o **"trasmessa su"**, o **"contenuta in qualcosa"**; questo **qualcosa** non è l'informazione stessa, ma il **supporto.**
- Ogni supporto ha le sue caratteristiche in quanto mezzo su cui può essere "scritta" dell'informazione.
  - Alcuni supporti sono adatti alla trasmissione ma non alla memorizzazione dell'informazione (aria, cavi,..) e viceversa (CD, hard disc,..).

Stessa informazione, diversi supporti





Stesso supporto, diversa informazione

tariffa, prezzo.

to make, to do, to build.

asino, cavalletto, somaro.

mantequilla, manteca.

Entità logiche e fisiche

- Distinguere informazione e supporto fisico è distinguere tra **entità logiche** ed **entità fisiche.**
  - L'informazione richiede un supporto fisico, ma non coincide con esso.
  - L'informazione è una entità extra-fisica, non interpretabile in termini di materia-energia e sottoposta alle leggi della fisica solo perché basata su un supporto fisico.

- **L'informazione si può creare e distruggere.**

Quando un sistema fisico supporta informazione?

- Si ottiene informazione quando, dato un insieme di alternative possibili, la lettura del supporto ne elimina alcune e ne seleziona altre.

- **Condizione necessaria** perché un supporto possa portare informazione è che possa assumere **configurazioni differenti**, a ognuna delle quali venga associata una differente **entità di informazione.**

Prima condizione sul supporto

- Un supporto che possa presentarsi sempre e comunque in un unico modo **non può** portare alcuna informazione.
- ***"Il supporto fisico deve consentire di distinguere tra le varie configurazioni attraverso determinate differenze."***
- Il caso più semplice è quello in cui **le configurazioni** del supporto sono **due.**

Configurazioni e codici

- A ogni configurazione del supporto deve essere associata una entità di informazione. A esempio:
  - interruttore premuto = "luce accesa"
  - interruttore rilasciato = "luce spenta".
- Per interpretare le differenti configurazioni del supporto in termini di informazione è necessario conoscere il **codice**, ovvero la "regola" che ad ogni configurazione (ammessa) del supporto associa una entità di informazione.
- La definizione di un codice comporta che sia identificato in modo non ambiguo l'insieme delle **possibili configurazioni del supporto** e delle **possibili entità di informazione** cui ci si vuole riferire.
- Ad uno stesso supporto fisico possono essere associati più codici.

Possibili rappresentazioni dei numeri

- *Unaria*: una barretta per unita'
  - Ad es: IIIIIIIIIIIIIII rappresenta 15!!!

- *Numeri Romani* con multipli di 5:

  Alfabeto: I(uno) V(5) X(10) L(50) C(100) D(500) M(1000)

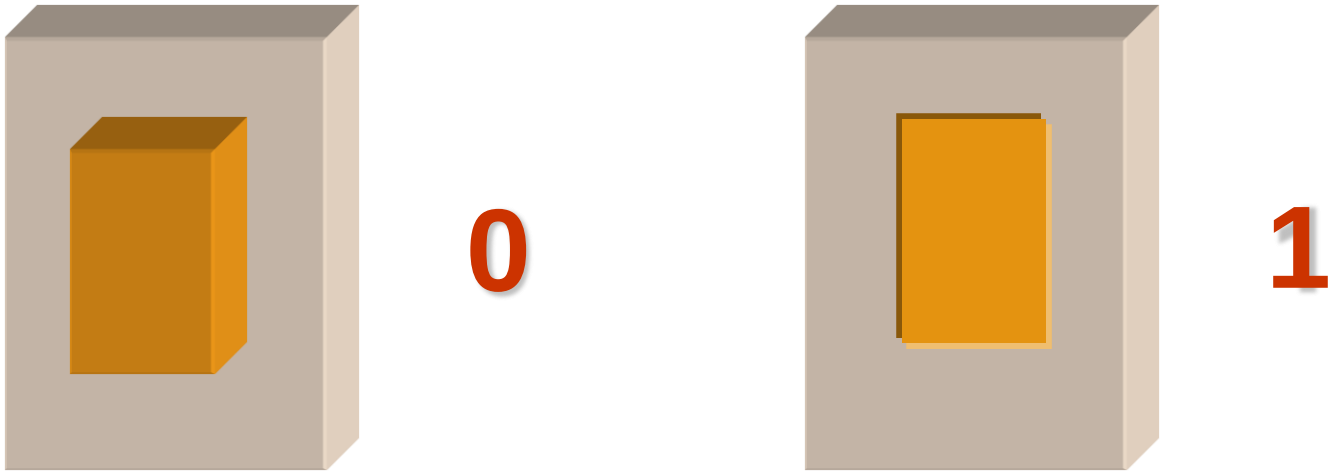
  Si *minimizza* numero di simboli ad es:
  I II III IV V …. XL (40) …. MCCLXXI (1271)….

  Problema con numeri > 4000:
  - Ad es. MMMMMMMMMMMM…

  Difficile eseguire le operazioni!

Il codice binario (1)

- Il **bit** è il supporto più semplice.
- Possiamo immaginare il bit come un **interruttore** che ha soltanto **due** posizioni (configurazioni):



Il codice binario (2)

- Definire un codice binario significa associare ad ogni configurazione di bit una certa entità di informazione.
- Anche se tipicamente le entità di informazione associate sono numeri **decimali**, è possibile associare qualsiasi insieme di oggetti all'insieme di configurazioni.
- La codifica binaria più semplice è quella ad **1 bit**, ovvero:

Il codice binario (3)

Si possono definire codifiche costituite da un numero **n** arbitrario di bit.

Ad esempio:

Codifica a 2 bit (4 configurazioni)

bin	dec
00	**0**
01	**1**
10	**2**
11	**3**

Codifica a 3 bit (8 configurazioni)

bin	dec
000	**0**
001	**1**
010	**2**
011	**3**
100	**4**
101	**5**
110	**6**
111	**7**

Il codice binario (4)

- Dato una **parola** di **n** bit, il numero delle possibili configurazioni è 2^n. Ad esempio, nella codifica a **3** bit vi sono $2^3 = 8$ configurazioni.
- Il codice binario è detto **posizionale**, in quanto ogni bit assume valore più o meno **significativo** a seconda della sua **posizione**. Tipicamente, più i bit sono posizionati verso sinistra, maggiore è il loro valore.
- La traduzione da binario a decimale si effettua moltiplicando il valore 2^k per ogni bit (dove **k** è la posizione del bit all'interno della codifica, partendo da destra) e sommando tutti i valori ottenuti.

Notazione posizionale in base 10

- La rappresentazione di un numero intero in base 10 è una sequenza di cifre scelte fra

  0 1 2 3 4 5 6 7 8 9

  - es: 23, 118, 4

- Il valore di una rappresentazione è dato da

  - $23 = 2*10^1 + 3 * 10^0 = 20 + 3$
  - $118 = 1*10^2 + 1*10^1 + 8 * 10^0 = 100 + 10 + 8$

Notazione posizionale in base 10: limite nella rappresentazione

- Il massimo numero rappresentabile con **n** cifre è 99….9 (**n** volte 9, la cifra che vale di più), pari a 10^n-1
  - es: su tre cifre il massimo numero rappresentabile è 999 pari a 10^3-1 =1000-1

Notazione posizionale in base 2

- La rappresentazione di un numero intero in base 2 è una sequenza di cifre scelte fra

  0 1

  - es: 10, 110, 1
- Il valore di una rappresentazione è dato da
  - $10 = 1*2^1 + 0*2^0 = 2$
  - $110 = 1*2^2 + 1*2^1 + 0*2^0 = 4 + 2 + 0 = 6$
  - $1 = 1*2^0 = 1$

Conversione da base 10 a base 2

Dato un numero N si calcola la sua rappresentazione in base 2

Conversione per divisione :

- si divide ripetutamente N per 2
- i resti ottenuti nella divisioni presi nell'ordine inverso in cui sono stati calcolati formano la sua rappresentazione binaria

Conversione da base 10 a base 2

Come si converte N nella sua rappresentazione in base 2 usando il metodo della divisione

Es : *convertiamo il numero 13*

- *13 / 2 = quoziente 6 e resto 1*
- *6 / 2 = quoziente 3 e resto 0*
- *3 / 2 = quoziente 1 e resto 1*
- *1 / 2 = quoziente 0 e resto 1*

La rappresentazione binaria di 13 è 1101

Conversione da base 10 a base 2

Si calcolano i resti delle divisioni per due

18 : 2 =	9	resto 0
9 : 2 =	4	resto 1
4 : 2 =	2	resto 0
2 : 2 =	1	resto 0
1 : 2 =	0	resto 1

10010

137 : 2 =	68	resto 1
68 : 2 =	34	resto 0
34 : 2 =	17	resto 0
17 : 2 =	8	resto 1
8 : 2 =	4	resto 0
4 : 2 =	2	resto 0
2 : 2 =	1	resto 0
1 : 2 =	0	resto 1

10001001

Da binario a decimale, un esercizio

La base é **b = 2**.
Si possono utilizzare solo le cifre **0** e **1**.

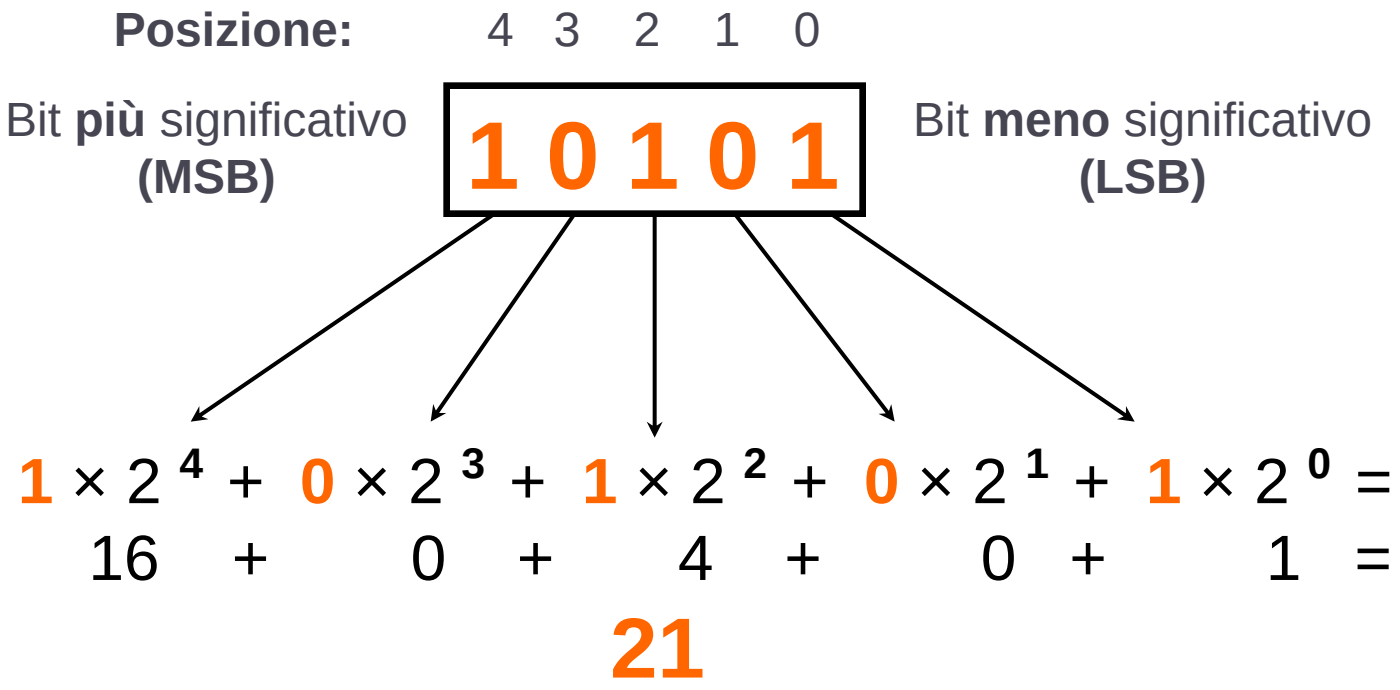
Esercizio

Trasformare il numero binario **101001011** nella corrispondente rappresentazione decimale.

$$\mathbf{101001011_2} = 1 \times 2^8 + 0 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 =$$
$$= (256 + 64 + 8 + 2 + 1) =$$
$$= \mathbf{331}$$

Un altro esempio

- Tradurre in decimale la seguente **parola** di 5 bit.



Notazione posizionale in base 2

- Per la base due valgono proprietà analoghe a quelle viste per la base 10
- Il massimo numero rappresentabile con N cifre è 11….1 (N volte 1, la cifra che vale di più), pari a 2^N-1
  - es: su tre cifre il massimo numero rappresentabile è 111 pari a $2^3-1 = 8 - 1 = 7$
- Quindi se voglio rappresentare K diverse configurazioni (cioè *0 1 2 …K-1*) mi servono almeno almeno *x* cifre dove 2^x è la più piccola potenza di 2 che supera K
  - es : se voglio 25 configurazioni diverse mi servono almeno 5 cifre perché $2^5=32$ è la più piccola potenza di 2 maggiore di 25

Base 2: un esercizio

Con **n** bit è possibile rappresentare **2^n** numeri naturali diversi, da **0** a **2^n-1**.

0 = 000 … 000 (n zeri)
2^n-1 = 111 … 111 (n uni)

Esercizio: quali sono i numeri naturali che si possono rappresentare con quattro bit ?

Soluzione: Tutti i numeri naturali compresi tra 0 e $2^4 - 1 = 16 - 1 = 15$

Rappresentazione dei numeri all'interno di un computer

- Usa la notazione binaria
- Ogni numero viene rappresentato con un numero finito di cifre binarie (*bit*)
- Numeri di *'tipo'* diverso hanno rappresentazioni diverse
  - Es. interi positivi, interi (pos. e neg.), razionali, reali, complessi
- Alcuni punti importanti:
  - se uso 4 byte (32 bit) posso rappresentare solo i numeri positivi da 0 a 232-1=231, che sono molti ma non *tutti*!
  - se moltiplico o sommo due numeri molto elevati posso ottenere un numero che non è rappresentabile
    - Es. vediamo cosa succede in base 10 con solo 3 cifre :

      500 + 636 = 1136 risultato 136

      se uso solo 3 cifre non ho lo spazio fisico per scrivere la prima cifra (1) che viene *'persa'*:
      Questo fenomeno viene chiamato *overflow*

Rappresentazione di interi all'interno di un computer

- Interi positivi e negativi:
  - es. il tipo `int` in C usa di solito 4 byte
  - ci sono diverse convenzioni di rappresentazione
    - Es: *modulo e segno* (MS) in cui il primo bit viene riservato al segno (1 negativo, 0 positivo) e gli altri 31 al modulo
  - rimane il problema dell'overflow

Rappresentazione di un insieme finito di oggetti (un esempio)

- Vogliamo rappresentare i giorni della settimana:

  {Lu, Ma, Me, Gio, Ve, Sa, Do}

  usando sequenze 0 e 1

- Questo significa costruire un *'codice'*, cioè una tabella di corrispondenza che ad ogni giorno associa una opportuna sequenza
- In principio possiamo scegliere la corrispondenza in modo del tutto arbitrario….

Rappresentazione di un insieme finito di oggetti (un esempio/1)

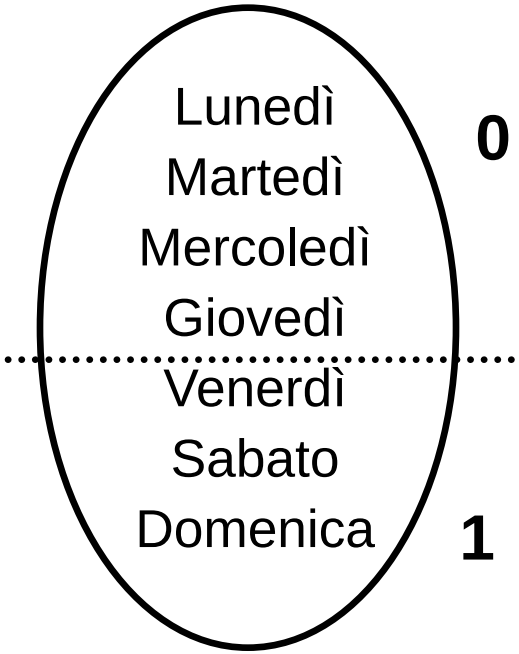
- Una possibile codifica binaria per i giorni della settimana
  - Lunedì 0100010001
  - Martedì 001
  - Mercoledì 1100000
  - Giovedì 1
  - Venerdì 101010
  - Sabato 111111
  - Domenica 000001
- Problema : la tabellina di corrispondenza fra codifiche tutte di lunghezza diversa
  - spreco di memoria
  - devo capire come interpretare una sequenza di codifiche
  - 11000001 1 = Me Gio Gio
  - 1 1000001 1 = Gio Gio Do Gio

Rappresentazione di un insieme finito di oggetti (un esempio/2)

- Di solito si usa un numero di bit uguale per tutti : il minimo indispensabile
- Per rappresentare 7 oggetti diversi servono almeno 3 bit (minima potenza di due che supera 7 è 8= 2^3) quindi:
  - 000 Lunedì
  - 001 Martedì
  - 010 Mercoledì
  - 011 Giovedì
  - 100 Venerdì
  - 101 Sabato
  - 110 Domenica
  - 111 *non ammesso*

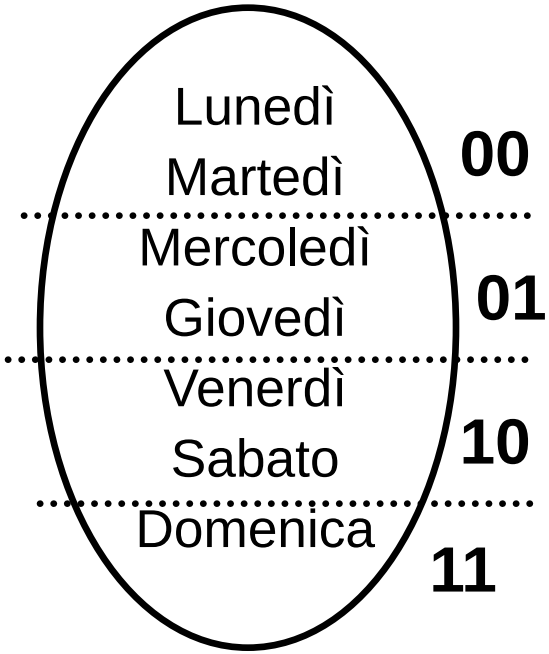
I giorni della settimana in binario

Codifica ad **1** bit



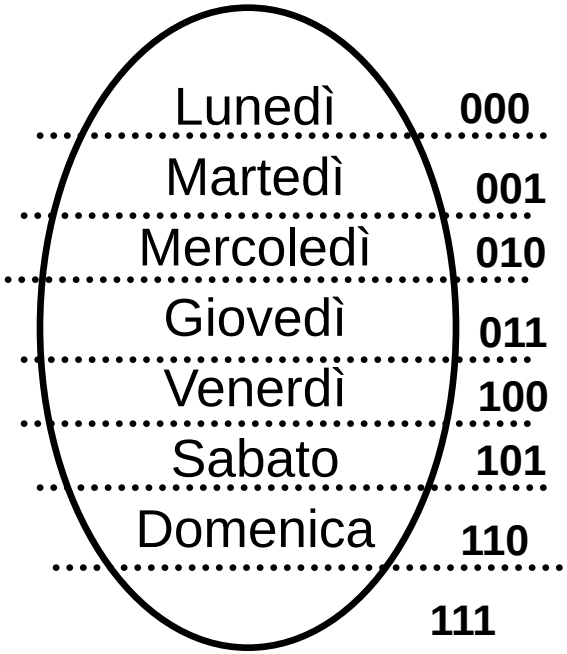
2 configurazioni

Codifica a **2** bit



4 configurazioni

Codifica a **3** bit



8 configurazioni