

Programmare in C++

A.S. 2015/2016

Alessandro Saltini

Liceo Scientifico Statale "A. Tassoni"

- ▶ Cos'è un computer?
 - ▶ logica binaria
 - ▶ bit come unità di informazione
 - ▶ numeri binari (ed hex?)
 - ▶ architettura di von Neumann
 - ▶ CPU (ALU/CU)
 - ▶ memorie (primarie / secondarie)
- ▶ Linguaggi
 - ▶ assembly (1-to-1 con machine code)
 - ▶ high-level languages
 - ▶ compilation process (preprocessor – compiler – linker)

- ▶ L'informatica **non** è
 - ▶ saper usare un computer
 - ▶ saper costruire/riparare un computer
 - ▶ usare programmi scritti da altri
- ▶ L'informatica **è**
 - ▶ una branca della matematica
 - ▶ lo studio dell'**informazione**
 - ▶ lo studio degli **algoritmi**
 - ▶ lo studio dei **linguaggi di programmazione**

- ▶ L'informazione si misura in **bit** (binary digit)
- ▶ 1 bit è la quantità di informazione necessaria a determinare una quantità che può essere **0** o **1**
- ▶ Il **byte** è un multiplo del bit: $1 \text{ B} = 8 \text{ bit}$
- ▶ Due scale di multipli del byte:
 - ▶ decimale: $\text{kB} (10^3)$, $\text{MB} (10^6)$, $\text{GB} (10^9)$, $\text{TB} (10^{12})$, ...
 - ▶ **binaria**: $\text{KiB} (2^{10})$, $\text{MiB} (2^{20})$, $\text{GiB} (2^{30})$, $\text{TiB} (2^{40})$, ...

- ▶ L'algebra Booleana è l'algebra dei bit
- ▶ È un **modello** della logica classica: 1 = vero, 0 = falso
- ▶ Insieme di base $B = \{ 0, 1 \}$
- ▶ Tre operazioni fondamentali:
 - ▶ **not** (non): $\neg : B \rightarrow B$
 - ▶ **and** (et): $\wedge : B^2 \rightarrow B$
 - ▶ **or** (vel): $\vee : B^2 \rightarrow B$

► not (non): \neg

► $\neg 1 = 0$

► $\neg 0 = 1$

► and (et): \wedge

► $1 \wedge 1 = 1$

► $1 \wedge 0 = 0$

► $0 \wedge 1 = 0$

► $0 \wedge 0 = 0$

► or (vel): \vee

► $1 \vee 1 = 1$

► $1 \vee 0 = 1$

► $0 \vee 1 = 1$

► $0 \vee 0 = 0$

- ▶ Combinando queste tre operazioni si possono ottenere tutte le altre operazioni possibili
- ▶ In realtà basta **una** sola operazione, meno intuitiva:
 - ▶ nand (\uparrow)
 - ▶ nor (\downarrow)
- ▶ Esistono circuiti **elettrici** che realizzano materialmente queste operazioni logiche
 - ▶ segnale "alto" = 1
 - ▶ segnale "basso" = 0
- ▶ Sono l'elemento di base dei computer

- ▶ Un numero in rappresentazione **decimale** è espresso come combinazione di potenze di 10

$$1064 = 1 \cdot 10^3 + 0 \cdot 10^2 + 6 \cdot 10^1 + 4 \cdot 10^0$$

- ▶ I coefficienti sono compresi tra 0 e 9 (**minori** di 10)
- ▶ Il massimo numero con n cifre decimali è $10^n - 1$
- ▶ I numeri esistono indipendentemente dalla loro rappresentazione, è solo un modo di scriverli

- ▶ La rappresentazione **binaria** utilizza le potenze di 2

$$10110 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$

- ▶ I coefficienti sono soltanto 0 e 1 (**minori** di 2)

- ▶ Il massimo numero con n cifre binarie è $2^n - 1$

- ▶ Ogni cifra è rappresentabile da un **bit**

- ▶ n cifre $\Rightarrow n$ bit

- ▶ I computer memorizzano i numeri in binario
- ▶ Le operazioni tra essi vengono svolte da appositi circuiti, basati sulle operazioni Booleane
- ▶ Ogni operazione richiede un certo tempo
- ▶ Limiti di memoria/tempo impediscono di operare con numeri arbitrariamente grandi

- ▶ I numeri negativi devono memorizzare anche il segno
- ▶ Costo di 1 bit aggiuntivo
 - ▶ $s = 0 \Rightarrow +$
 - ▶ $s = 1 \Rightarrow -$
- ▶ Spesso si ricorre a rappresentazioni alternative
 - ▶ rimozione di ambiguità tra $+0$ e -0
 - ▶ facilità di calcolo
 - ▶ occupano comunque 1 bit in più

- ▶ La parte frazionaria è problematica da rappresentare

$$1.1011 = 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4}$$

- ▶ Non tutti i numeri con rappresentazione decimale finita hanno rappresentazione binaria finita
- ▶ Non possiamo memorizzare infinite cifre
 - ▶ Impossibile rappresentare i numeri irrazionali
 - ▶ Non tutti i numeri razionali sono rappresentabili

- ▶ Richiamiamo la notazione scientifica

$$1064.15 = 1.06415 \cdot 10^3$$

- ▶ Generalizzabile in binario come

$$110.1011 = 1.101011 \cdot 2^2$$

- ▶ La **prima** cifra della rappresentazione scientifica binaria è **sempre 1**, non serve memorizzarla

$$110.1011 = 1.101011 \cdot 2^2$$

- ▶ La parte dopo la virgola è detta **mantissa** o **significando**, è un numero intero
- ▶ L'**esponente** di 2 è un numero intero
- ▶ Un numero frazionario viene rappresentato come coppia di numeri interi
 - ▶ bit del significando \Rightarrow precisione
 - ▶ bit dell'esponente \Rightarrow range