

Università di Roma Tre – Dipartimento di Matematica e Fisica – Corso di Laurea in Matematica

Appunti del corso di Informatica 1 (IN110 – Fondamenti)

5 – Rappresentazione delle informazioni

Marco Liverani
(liverani@mat.uniroma3.it)

Sommario

- Cenni sulla numerazione in base 2, in base 8 ed in base 16
- Organizzazione della memoria del calcolatore in *bit* e *byte*
- Convenzioni per la rappresentazione di numeri interi, razionali, caratteri alfanumerici
- Il codice ASCII
- I principali tipi di dato
- I puntatori

Numerazione in base 10

- Nella vita di tutti i giorni siamo abituati a rappresentare i numeri in **base 10**
- Ogni numero viene rappresentato fattorizzandolo in multipli di potenze di 10:
- Es.: $365_{10} = 3 \times 10^2 + 6 \times 10^1 + 5 \times 10^0$
- In questo tipo di numerazione utilizziamo 10 simboli convenzionali (l'*alfabeto* della nostra codifica): 0, 1, 2, 3, ..., 9
- Utilizzando un criterio simile possiamo scegliere di rappresentare i numeri con basi diverse da 10, ad esempio la base 2 o la base 16

Numerazione in base 2

- La macchina opera con una logica *binaria* che riflette direttamente la struttura fisica delle sue componenti
- Nella numerazione binaria ogni numero viene rappresentato fattorizzandolo in multipli di potenze di 2:

$$\begin{aligned} 365_{10} &= 256 + 64 + 32 + 8 + 4 + 1 \\ &= 1 \times 2^8 + 0 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 \\ &\quad + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 101101101_2 \end{aligned}$$

Numerazione in base 16

- Nella numerazione in base 2 l'alfabeto dei simboli che possiamo utilizzare per rappresentare i numeri è composto da {0, 1}
- In base 16 (numerazione **esadecimale**) possiamo utilizzare 16 simboli: {0, 1, 2, ..., 9, A, B, ..., F}
- A rappresenta il numero 10 ($A_{16}=10_{10}$), B l'11, C il 12, D il 13, E il 14 ed F il 15. Il numero 16_{10} è rappresentato da 10_{16} , ossia

$$16_{10} = 1 \times 16^1 + 0 \times 16^0 = 10_{16}$$

Informazioni numeriche

1

- La memoria della macchina è un "casellario" molto grande suddiviso in **locazioni di memoria**, numerate progressivamente mediante degli **indirizzi di memoria** che ne identificano univocamente la posizione
- Ogni locazione è composta da un insieme di **8 bit** che compongono un **byte**
- Con un solo byte è possibile rappresentare piccoli numeri interi (compresi tra $0_{10}=00000000_2$ e $255_{10}=11111111_2$)
- Per rappresentare numeri più grandi la macchina aggrega più locazioni di memoria contigue: ad esempio con 2 byte è possibile rappresentare numeri binari di 16 cifre (16 bit), compresi tra 0 e 65.535

Informazioni numeriche

2

- Per rappresentare **numeri con il segno** (interi relativi) si adotta la *convenzione* di considerare il primo bit come rappresentante del segno: ad esempio 0 per il segno negativo ed 1 per il segno positivo
- Con 2 byte (16 bit, di cui 15 per la rappresentazione del numero ed 1 per il segno) potremo così rappresentare numeri compresi tra $+32.767$ e -32.767
- Complessivamente vengono così rappresentati comunque 65.535 numeri, di cui 32.768 positivi (considerando anche lo zero) e 32.767 negativi

Informazioni numeriche

3

- Per rappresentare **numeri con la virgola** (razionali positivi o negativi) si utilizza la notazione scientifica:
$$-12,345 = -12345 \times 10^{-3}$$
- Quindi basta adottare un'altra convenzione: ad esempio, su un insieme di 32 bit il primo rappresenterà il segno, con 28 bit rappresenteremo le cifre significative del numero e con gli ultimi 3 bit rappresenteremo l'esponente negativo
- Così si possono rappresentare **alcuni** numeri razionali compresi, tra $-268.435.455$ e $+268.435.455$, con un massimo di 8 cifre dopo la virgola. Più è grande (in valore assoluto) il numero e meno cifre decimali potremo rappresentare

Informazioni non numeriche

- Con i computer spesso si trattano informazioni non numeriche, come caratteri alfabetici (o meglio, alfanumerici) o rappresentazioni grafiche.
- Mediante opportune *convenzioni* è possibile rappresentare utilizzando la codifica binaria ogni tipo di informazione.
- Per i **caratteri alfanumerici** (caratteri alfabetici, simboli di interpunzione, cifre numeriche ed altri simboli ancora) esiste una tabella di codifica standard che associa ad ogni carattere un codice numerico intero: la **codifica ASCII** (*American Standard Code for Information Interchange*).
- Ad esempio il carattere “a” è associato al codice 61, al carattere “b” il 62, e così via.

Tipi di dato e dichiarazione di variabili

1

- Una medesima sequenza di bit può dunque rappresentare un numero intero positivo, un differente numero relativo, un numero razionale ovvero un carattere alfanumerico.
- Per indicare alla macchina come dovrà essere trattata una certa sequenza di bit memorizzati in un determinato blocco della memoria, è necessario che il programmatore a priori **dichiari il tipo di dato** che intenderà associare ad una certa variabile nell’ambito di un intero programma o di una determinata funzione.
- Con la dichiarazione del tipo di una variabile si indica anche alla macchina la **quantità di memoria** che dovrà essere riservata (**allocata**) per la memorizzazione delle informazioni trattate dal programma.

Tipi di dato e dichiarazione di variabili

2

- In ogni linguaggio di programmazione vengono messi a disposizione del programmatore dei **tipi di dato elementari** con cui possono essere definite le **variabili** o le **strutture dati** più complesse ed articolate.
- I tipi di dato fondamentali, disponibili in quasi tutti i linguaggi di programmazione sono i seguenti:
 - **Intero** (in C: short, unsigned, int e long)
 - **Floating point** (in C: float, double)
 - **Carattere** (in C: char)
- I **puntatori** sono delle variabili che contengono l'**indirizzo di memoria** in cui è allocata un'altra variabile: si dice così che **puntano** ad un'altra variabile.