

# 12BHD INFORMATICA, A.A. 2017/2018

## Esercitazione di Laboratorio 4

---

### Obiettivi dell'esercitazione

- Risolvere problemi che implicino scelte logiche
- Sperimentare il concetto di iterazione

### Contenuti tecnici

- Uso dei costrutti *if-then-else* e *switch*
  - Uso dei costrutti iterativi *while*, *do-while* e *for*
  - Introduzione all'uso degli operatori *cast* e *sizeof*
- 

### Da risolvere preferibilmente in laboratorio

Esercizio 1. Si scriva un programma in linguaggio C in grado di determinare se l'equazione di secondo grado ( $ax^2 + bx + c = 0$ ) ha soluzioni reali.

In particolare:

- a. Si definiscano tre variabili chiamate  $a$ ,  $b$  e  $c$ , corrispondenti ai parametri dell'equazione
- b. Si acquisisca da tastiera il valore di  $a$ ,  $b$  e  $c$
- c. Si calcoli il cosiddetto *discriminante* della formula risolutiva
  - i. In caso il delta sia positivo, visualizzare il seguente messaggio "L'equazione ha due soluzioni REALI distinte"
  - ii. In caso il delta sia nullo, visualizzare il seguente messaggio "L'equazione ha due soluzioni REALI coincidenti"
  - iii. Altrimenti stampare a video un messaggio per segnalare che l'equazione non ha soluzioni reali

Esercizio 2. Si scriva un programma in linguaggio C che, dato un numero intero tra 1 e 12 che rappresenta il mese corrente, utilizzi il costrutto *switch* per stampare il nome del mese per esteso (1 → "Gennaio", 2 → "Febbraio", 3 → "Marzo", ..., 12 → "Dicembre").

Il programma gestisca anche le situazioni di inserimento di valori non compresi nell'intervallo 1-12.

Approfondimento: modificare il programma in modo che accetti come input una data nella forma gg/mese/anno (esempio: 23/3/2012) e stampi la stessa data con il mese per esteso (esempio: 23 marzo 2012). Si consiglia l'utilizzo del costrutto *switch*.

Esercizio 3. Si scriva un programma C che acquisisca numeri interi da tastiera finché non viene inserito il valore 0.

Suggerimento: si utilizzi il costrutto iterativo *while* oppure *do-while*.

Approfondimento: modificare il programma accumulando (ovvero continuando a sommare) in una variabile  $i$  valori inseriti prima dell'immissione del numero 0; al termine dell'acquisizione il programma stampi a video il valore calcolato.

Da risolvere a casa

Esercizio 4. Scrivere un programma C che acquisisca in input da tastiera un valore intero positivo  $N \leq 40$  corrispondente alla base di un triangolo rettangolo e isoscele, e che riproduca a video tale triangolo utilizzando il carattere ‘\*’.

Esempio: se il valore letto da tastiera è 3, a video dovrà essere visualizzata la seguente serie di caratteri:

```
*
**
***
```

Approfondimento: Si scriva un programma in linguaggio C che letto un valore intero positivo dispari  $N$  disegni forme geometriche alternative, quali il triangolo isoscele, il quadrato... ecc.

Per esempio, provare a disegnare la seguente figura geometrica:

```
*****
 *
 *
 *
 *
 *
 *
 *
```

In questo caso il valore  $N = 9$ .

Esercizio 5. Scrivere un programma in linguaggio C che visualizzi i primi 20 numeri della serie di Fibonacci.

Suggerimento: ecco i primi numeri appartenenti alla serie 0 1 1 2 3 5 8 ... In modo formale la serie si costruisce considerando la seguente relazione:

$X_i = X_{i-1} + X_{i-2}$ , con  $X_0 = 0$  e  $X_1 = 1$ ;

Approfondimento: si modifichi la serie come segue:

$X_i = X_{i-1} * X_{i-2}$ , con  $X_0 = 1$  e  $X_1 = 2$ ; quanti sono gli elementi di questa serie rappresentabili con variabili di tipo intero?

Esercizio 6. <sup>1</sup>Scrivere un programma C che legga in input da tastiera un numero reale  $N$  e successivamente numeri interi finché entrambe le seguenti condizioni sono rispettate

- La media dei numeri acquisiti è superiore al valore di  $N$
- Sono stati acquisiti meno di 10 numeri.

Esercizio 7. Si scriva un programma C con lo scopo di calcolare il massimo valore positivo memorizzabile in variabili di tipo *int*, *long* e *unsigned int*.

Suggerimento: seguendo la traccia riportata nel seguito, eseguire le istruzioni del programma con il passo-passo del debug e osservare il risultato delle varie assegnazioni.

a) Verificare che non è una via percorribile provare ad assegnare valori via via più grandi: se ad esempio si scrive l’istruzione *value = 3000000000*, il compilatore non segnala errore, e al più segnala warning. Cosa si osserva in *value* con il watch, dopo l’esecuzione dell’istruzione?

b) Provare a ottenere questi valori in modo “empirico”, ovvero acquisendoli tramite la funzione *scanf* e ristampandoli opportunamente usando la *printf*. Verificare tramite il pro-

---

<sup>1</sup> Questo esercizio sarà svolto in modo multimediale e inserito sul Portale, tra il materiale comune, nelle settimane successive.

gramma che anche questa via non è percorribile: il comportamento della *scanf* in caso di errore nei dati non è dominabile da chi scrive il programma.

c) Realizzare a questo punto un algoritmo che, tenendo conto delle rappresentazioni binarie dei numeri senza segno e in complemento a 2, permetta di rilevare il valore max: per i numeri con segno, si può attribuire a *value* il valore iniziale di 0, poi si incrementa ripetutamente *value*. È noto che se si incrementa di 1 il valore massimo positivo, si ottiene overflow e il valore diventa negativo. Il valore cercato è dunque il precedente al primo valore negativo trovato. Tradurre l'algoritmo in programma e collaudarlo. Come si può modificare l'algoritmo (e il programma) perché operi con i numeri senza segno?