

# Esercitazione di Laboratorio 01

---

## Temi trattati

---

1. Primi approcci al pensiero computazionale
  - a. Computer e programmi informatici
  - b. Problemi ben formulati e algoritmi
  - c. Diagrammi di flusso e pseudocodice
2. Primi passi con il linguaggio Python
  - a. Set-up dell'ambiente di sviluppo
  - b. Creazione ed esecuzione programmi
  - c. Stampe a terminale

## Discussione

---

- A. Qual è la differenza tra utilizzare un programma informatico e programmare un computer?
- B. Un tostapane svolge un solo compito. Un computer può essere programmato per svolgere compiti diversi. Le macchine seguenti hanno funzioni singole o sono computer programmabili?
  - a. Smartphone
  - b. Lavatrice
  - c. Juke-box
  - d. Reflex digitale
- C. Quali caratteristiche deve avere un algoritmo per essere compreso ed eseguito da un computer?
- D. In quale relazione sono il problema di partenza, lo pseudocodice, il programma informatico, il diagramma di flusso e il linguaggio di programmazione?

## Esercizi

---

### Parte 1 - Dal problema all'algoritmo

**Consegna:** valutare se i problemi seguenti sono posti nel modo adatto ad essere gestiti da un programma informatico. In caso positivo, dove non fornito nel testo dell'esercizio, descrivere un algoritmo (in forma di pseudocodice e/o diagramma di flusso) per risolverli. In caso negativo, scrivete il motivo per cui non possono essere gestiti. Completare almeno i primi quattro esercizi durante l'esercitazione, e i rimanenti a casa. Non è richiesto di scrivere il codice Python.

**01.1.1 Appuntamenti.** In un programma di calendarizzazione (scheduling) di eventi, verificare se due appuntamenti della stessa giornata si sovrappongono. Per semplicità, ipotizziamo che gli appuntamenti inizino sempre a un'ora esatta (senza minuti) e usiamo l'orario militare (cioè con le ore che vanno da 0 a 23). Determinare se l'appuntamento che inizia all'ora *start1* e termina all'ora *end1* si sovrappone all'appuntamento che inizia all'ora *start2* e termina all'ora *end2*. Lo pseudocodice seguente descrive un possibile algoritmo per determinare una sovrapposizione.

```
Se start1 > start2
    s = start1
Altrimenti
    s = start2
Se end1 < end2
```

```

        e = end1
    Altrimenti
        e = end2
    Se s < e
        Gli appuntamenti si sovrappongono
    Altrimenti
        Gli appuntamenti non si sovrappongono

```

Seguite passo dopo passo l'esecuzione dello pseudo-codice con gli appuntamenti 10–12 e 11–13, poi con gli appuntamenti 10–11 e 12–13; disegnate il diagramma di flusso per l'algoritmo; spiegate il funzionamento dell'algoritmo e verificate se esso sia corretto; spiegate il significato delle variabili s ed e. [R3.12]

**01.1.2 Stagioni.** L'algoritmo seguente individua la stagione (Spring, Summer, Fall o Winter, cioè, rispettivamente, primavera, estate, autunno o inverno) a cui appartiene una data, fornita come mese e giorno.

```

    Se mese è 1, 2 o 3
        stagione = "Winter"
    Altrimenti se mese è 4, 5 o 6
        stagione = "Spring"
    Altrimenti se mese è 7, 8 o 9
        stagione = "Summer"
    Altrimenti se mese è 10, 11 o 12
        stagione = "Fall"
    Se mese è divisibile per 3 e giorno >= 21
        Se stagione è "Winter"
            stagione = "Spring"
        Altrimenti se stagione è "Spring"
            stagione = "Summer"
        Altrimenti se stagione è "Summer"
            stagione = "Fall"
    Altrimenti
        stagione = "Winter"

```

Disegnate il diagramma di flusso per l'algoritmo. Verificate se l'algoritmo si comporta correttamente con una serie di date di prova. [R3.13]

**01.1.3 Sulla strada.** Calcolare la percentuale di utilizzo della vostra automobile per uso personale e, separatamente, per recarvi al lavoro. Conoscete la distanza tra casa vostra e il vostro luogo di lavoro e, dato un periodo di tempo, avete registrato il valore riportato dal contachilometri all'inizio ed alla fine del periodo; inoltre è noto il numero di giorni in cui vi siete recati al lavoro in tale periodo. [R1.16]

**01.1.4 Decorazioni.** Descrivere come tassellare a scacchiera il pavimento di una stanza a base rettangolare, i cui lati misurano entrambi multipli di 10 cm, con piastrelle bianche e nere quadrate di misura 10 x 10 cm. [Worked Example 1.1]

**01.1.5 Decisioni future.** Una persona alla fine delle scuole superiori vuole decidere a quale facoltà universitaria iscriversi, in base ai consigli che riceve e allo stipendio percepito il primo anno di lavoro dopo la laurea. Per la maggior parte delle facoltà sono disponibili stime occupazionali dei neolaureati. Come può decidere?

**01.1.6 Sapere quando fermarsi.** Il valore del pi greco può essere calcolato con la formula seguente:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

Come è possibile ottenere un valore che approssima pi greco? Dal momento che l'algoritmo avrebbe istruzioni infinite di per sé, imporre che il risultato debba avere sei cifre dopo la virgola. [R1.18]

**01.1.7 Tornare alle radici.** Si consideri il seguente algoritmo di tipo matematico:

Dato un valore  $\alpha > 0$ , un algoritmo per approssimare  $\sqrt{\alpha}$  comunemente usato è conosciuto come metodo babilonese e sfrutta gli stessi principi poi codificati nel [metodo di Newton](#). Questo metodo funziona nel modo seguente:

1. Poni  $n = 1$  e inizia con un valore arbitrario positivo  $x_n$  (quanto più esso è prossimo alla radice, tanto migliore è la convergenza dell'algoritmo)
2. sostituisci  $x_n$  con la media di  $x_n$  e  $\alpha/x_n$
3. aumenta  $n$  e vai al punto 2

Questo algoritmo può essere rappresentato da

$$x_{n+1} = \frac{1}{2} \left( x_n + \frac{\alpha}{x_n} \right)$$

da cui si ricava  $\lim_{n \rightarrow \infty} x_n = \sqrt{\alpha}$ .

Utilizzare tale algoritmo per calcolare  $\sqrt{3}$ .

## Parte 2 – Dall'algoritmo al programma informatico

**Consegna:** per ciascuno degli esercizi seguenti, scrivere un programma in Python che stampi a video gli elementi richiesti. Se c'è tempo a sufficienza, completare gli esercizi durante l'esercitazione, altrimenti completarli a casa.

**01.2.1** Una frase di inaugurazione delle esercitazioni del corso di informatica. [P1.1]

**01.2.2** La somma dei primi 10 numeri interi positivi:  $1 + 2 + \dots + 10$ . [P1.2]

**01.2.3** Il prodotto dei primi 10 numeri interi positivi:  $1 \times 2 \times \dots \times 10$ . [P1.3]

**01.2.4** Una parola a scelta con lettere grandi, come ad esempio:

```

AAA      SSSSSS  CCCCCC  IIII  IIII
AA AA    SS     SS CC   CC   II   II
AA  AA   SS     CC           II   II
AA    AA  SSSSSS CC           II   II
AAAAAAAA  SS CC           II   II
AA    AA SS   SS CC   CC   II   II
AA    AA  SSSSSS CCCCCC  IIII  IIII

```

[P1.6]

**01.2.5** Il vostro nome, in colonna.

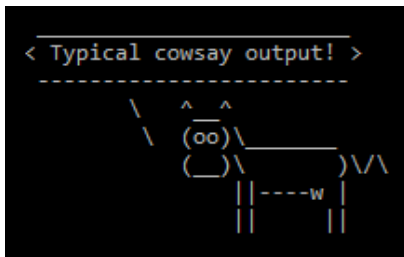
**01.2.6** Il saldo di un conto bancario dopo il primo, secondo e terzo anno. Il conto ha un saldo iniziale di 1000 dollari e vi vengono accreditati interessi annuali al 5%. [P1.4]

**01.2.7** Il vostro nome all'interno di un rettangolo, utilizzando i caratteri: `|` `-` `+`, come nell'esempio seguente:

```
+-----+
|  Dave  |
+-----+
```

[P1.5]

**01.2.8** Un animale a vostra scelta con un fumetto contenente un saluto, senza usare il comando [cowsay](#).



[P1.10]

**01.2.9** Un codice alfanumerico lungo 16 caratteri che alterni le stringhe `"abcd"` e `"1234"`.

**01.2.10** Una scacchiera di dimensioni 5x5 in cui le caselle bianche sono rappresentate da `"0"`, e le caselle nere da `"1"`.

**01.2.11** Una linea composta da 100 trattini `"-"`.

**01.2.12** Una sequenza di cifre composta da 100 zeri.

**01.2.13** Il quarto elemento della sequenza di Fibonacci, dove ciascun elemento ammonta alla somma dei due precedenti.

**01.2.14** I primi quattro elementi della sequenza di Fibonacci, in colonna.

**01.2.15** Una frase di chiusura di questa prima esercitazione, all'interno di una cornice grafica a vostra scelta, che includa il calcolo della percentuale di esercizi che avete portato a compimento.