

# Esercitazione 1 - Laboratorio

Informatica@DSS 2022/2023

Massimo Lauria <massimo.lauria@uniroma1.it>\*

Lo scopo del laboratorio è di esercitarsi e misurare la propria preparazione: gli esercizi non sono troppo difficili, se si sono seguite le lezioni. Non vi viene comunque messo alcun voto.

**Modalità di lavoro:** gli studenti devono lavorare in autonomia o in piccoli gruppi, senza disturbare i colleghi. Il lavoro di gruppo è fruttuoso solo se tutti partecipano e se ognuno scrive una propria soluzione per tutti gli esercizi.

Il docente cercherà per quanto possibile di non occupare il tempo del laboratorio per introdurre materiale nuovo, anche se a volte questo sarà necessario. Il docente è a disposizione per aiutare gli studenti, che possono iniziare a lavorare anche prima che il docente arrivi in aula, se lo desiderano

**Raccomandazioni** leggete bene il testo degli esercizi prima di chiedere chiarimenti. In ogni caso sarò in aula con voi.

## 1 Input da tastiera

Usate le funzioni `input` e `float` per leggere due numeri da tastiera, e stampate la somma su schermo usando `print`

Ripetete poi lo stesso esercizio, ma i due dati letti da `input` devono essere due stringhe e le due stringhe devono essere concatenate. Cosa va cambiato nel programma precedente?

---

\*<http://massimolauria.net/informatica2022/>

## 2 Sconto

Scrivete un programma che calcoli il prezzo scontato. Ad esempio se il prezzo è 200 euro e lo sconto è il 15% il programma deve stampare il valore 170. Il programma dovrà

- leggere da tastiera il prezzo originale e lo sconto
- verificare che lo sconto sia valido (ovvero un numero tra 0 e 100)
- verificare che il prezzo sia un numero non negativo (ovvero maggiore o uguale a 0)

Se una di queste condizioni non si dovesse verificare, il programma deve stampare la stringa 'errore' per indicare che gli argomenti passati non sono validi.

## 3 Numero float più piccolo

I float sono rappresentazioni dei numeri reali che hanno una precisione limitata, a causa del fatto che ogni float occupa una quantità di memoria fissa (al contrario degli int che possono crescere a piacere e quindi occupare una quantità di memoria potenzialmente illimitata).

Entrate nell'interprete interattivo e cercate di capire quale sia il numero float più piccolo che possa essere rappresentato tra quelli **maggiori di zero**, e che però non sia approssimato a zero da python.

*Suggerimento:* il modo più rapido per costruire numeri piccoli è quello di prendere un numero strettamente compreso tra 0 e 1 ed elevarlo a potenza. Più la potenza cresce più il numero si rimpicciolisce, ma non è mai zero. Ad un certo punto può darsi però che python non sia più in grado di rappresentare un numero così piccolo e lo consideri zero.

## 4 Equazione di secondo grado

Scrivete un programma che calcoli e restituisca la soluzione **più grande**, tra quelle dell'equazione di secondo grado  $Ax^2 + Bx + C = 0$ .

Se l'equazione non ha soluzione allora il programma deve stampare la stringa 'nessuna soluzione'. Altrimenti deve restituire il valore numerico.

Il programma legge i valori  $A, B, C$  da tastiera.

*Suggerimento:* ricordate che se  $A$  è zero dovrete trattare l'equazione come se fosse di primo grado. Per calcolare la radice quadrata potete fare elevare ad  $1/2$ . Ad esempio  $12^{0.5}$ .

## 5 Espressione

Quanto vale l'espressione seguente?

```
not -5//2**4 < -1 and 3 ** 2 ** (5 + - 3) >= 2*4
```

1

Per scoprirlo può essere utile

1. capire quali siano le precedenze degli operatori;
2. costruire il diagramma ad albero che indichi la struttura della valutazione dell'espressione, come abbiamo visto in aula;
3. calcolare il valore che corrisponda ad ogni sottoespressione dal basso verso l'altro;
4. Usare Thonny e le funzioni Debug / Step-Into per valutare l'espressione.

## 6 Ordinare tre valori (numerici?)

Scrivete una programma che stampi tre valori passati come argomenti in ordine dal più piccolo al più grande. Ad esempio se i tre valori sono 5.2, 1.4, 2.2 dovrebbe stampare nell'ordine 1.4, 2.2, e 5.2.

Per ora non vi preoccupate di gestire gli errori dovuti al passaggio di argomenti che non sono confrontabili tra loro.

Il programma deve leggere tre valori  $A, B, C$  da tastiera, e poi deve stamparli in ordine crescente.

## 7 Area del cerchio e Volume del cilindro

In modo analogo agli esercizi precedenti, scrivete due programmi `areacerchio` e `volumecilindro`.

- Quanti e quali parametri deve leggere da tastiera `areacerchio`?

- Quanti e quali parametri deve leggere da tastiera volumecilindro?

## 8 Programmi più comodi per l'utente

Attraverso l'uso delle funzioni `input` (alla quale potete passare un argomento opzionale) e `print`, potete rendere un po' più comodi da usare tutti programmi che avete scritto oggi. L'utente in linea di principio non può o non vuole vedere il codice Python, e quindi deve essere messo in condizione di capire cosa deve fare per usare il programma.

- Fate capire all'utente quando deve inserire dei dati da tastiera.
- Fate capire quale dato sta inserendo di volta in volta.
- In caso di errori riconosciuti, stampate dei messaggi chiari appropriati, e in ogni caso non stampate risultati non corretti.
- In caso di input validi i programmi devono comunicare il risultato con frasi di senso compiuto.