

Esercitazione di Laboratorio 02

Temi trattati

1. Le variabili in Python
 - a. I tipi di variabile
 - b. Gli operatori aritmetici di base
 - c. La manipolazione di stringhe

Discussione

- A. In quale relazione sono il valore assegnato, la variabile e il tipo di dato?
- B. Fare un esempio di operazioni attuabili su variabili con valori assegnati corrispondenti a diversi tipi di dato.
- C. Consideriamo l'operatore "+". Cosa succede se viene utilizzato tra...
 - a. Due variabili con valori di tipo `int`;
 - b. Una variabile con valore di tipo `int` e una con valore di tipo `float`;
 - c. Due variabili con valori di tipo `str`;
 - d. Una variabile con valore di tipo `int` e una con valore di tipo `str`.

Esercizi

Parte 1 - Operazioni aritmetiche

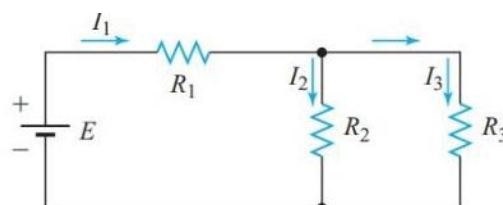
Consegna: per ciascuno degli esercizi seguenti, scrivere un programma in Python che risponda alle richieste indicate. Completare almeno due esercizi durante l'esercitazione, e i rimanenti a casa.

02.1.1 Due numeri. Scrivere un programma che memorizzi due numeri interi in due costanti definite nel codice, e poi ne visualizzi:

- A. La somma;
- B. La differenza;
- C. Il prodotto;
- D. Il valore medio;
- E. La distanza (cioè il valore assoluto della differenza);
- F. Il valore massimo (cioè il maggiore tra i due);
- G. Il valore minimo (cioè il minore tra i due).

Suggerimento: utilizzare le funzioni `max()` e `min()` definite in Python. Esse accettano una sequenza di valori separati da virgola in input e restituiscono rispettivamente il valore massimo e minimo della sequenza (ad esempio, `max(10, 5)` restituisce `10`). [P2.4]

02.1.2 Resistenze. Considerare il circuito seguente.



Scrivere un programma che, a partire dalle resistenze dei tre resistori, immesse in input dall'utente, calcoli la resistenza totale, utilizzando la legge di Ohm. [P2.38]

02.1.3 Cifre. Scrivere un programma che memorizzi in una costante un numero intero positivo di cinque cifre (quindi compreso tra 10000 e 99999), e visualizzi le singole cifre di cui è composto. Ad esempio, avendo il numero **16384**, il programma deve visualizzare, su righe separate: **1 6 3 8 4**. [P2.16]

02.1.4 Auto ibrida. Scrivere lo pseudocodice e il relativo programma Python che aiuta una persona a decidere se comprare o meno un'auto ibrida. Gli input del programma dovrebbero essere:

- I. il costo di un'auto nuova;
- II. la stima dei chilometri percorsi in un anno;
- III. La stima del costo del carburante;
- IV. L'efficienza in chilometri al litro;
- V. La stima del valore di rivendita dell'auto usata dopo 5 anni.

Calcolare il costo totale di proprietà dell'auto per 5 anni (per semplicità, non tenere in considerazione il costo di finanziamenti). Per fornire gli input al programma, ricercare sul Web prezzi e consumi realistici per due alternative per l'acquisto di un'auto nuova nella stessa fascia di prezzo: un modello ibrido e uno a benzina. Eseguire il programma due volte per comparare le due alternative, considerando l'attuale prezzo del carburante e la previsione di percorrere 30'000 chilometri all'anno. [P2.10]

02.1.5 Forza elettrica. Secondo la legge di Coulomb, la forza elettrica (espressa in Newton) tra due particelle cariche con carica, rispettivamente, Q_1 e Q_2 , separate da una distanza r , è

$$F = \frac{Q_1 Q_2}{4 \pi \epsilon r^2}$$

dove $\epsilon = 8.854 \times 10^{-12}$ Farad / metro. Scrivere un programma che calcoli e mostri a video la forza relativa ad una coppia di particelle cariche, permettendo all'utente di scegliere i valori Q_1 , Q_2 (in Coulomb) e r (in metri). [P2.43]

Parte 2 – Manipolazione di stringhe

Consegna: per ciascuno degli esercizi seguenti, scrivere un programma in Python che risponda alle richieste indicate. Completare almeno due esercizi durante l'esercitazione, e i rimanenti a casa.

02.2.1 Caratteri. Scrivere un programma che memorizzi una stringa in una variabile e, a partire da quella variabile, visualizzi i primi tre caratteri della stringa, seguiti da tre punti, ancora seguiti dagli ultimi tre caratteri. Ad esempio, se la stringa viene inizializzata al valore "**Mississippi**", il programma deve visualizzare "**Mis...ppi**". Cosa succede della stringa è più corta di 6 caratteri? E se è più breve di 3 caratteri? [P2.22]

02.2.2 Numero telefonico. Lo pseudocodice seguente descrive come trasformare una stringa contenente un numero telefonico a dieci cifre (come "**4155551212**") in una stringa più facilmente leggibile, formattata secondo lo stile statunitense, come "**(415) 555-1212**":

- I. Prendere la stringa costituita dai primi tre caratteri e circondarla con parentesi tonde (questo è il prefisso, detto "area code");
- II. Concatenare il prefisso con la stringa contenente i tre caratteri successivi, un trattino e la stringa costituita dagli ultimi quattro caratteri si ottiene il numero nel formato richiesto.

Tradurre questo pseudocodice in un programma Python che memorizzi un numero telefonico di 10 cifre in una stringa, per poi visualizzarlo nel formato appena descritto. [P2.33]

02.2.3 Allineamento. Formattare l'output dell'esercizio **02.1.1** in modo che descrizioni e numeri siano allineati verticalmente, ad esempio:

```
Somma:      45
Differenza: -5
...         ...
```

[P2.5]

02.2.4 Emoji. Secondo i dati raccolti dall'[Unicode Consortium](https://home.unicode.org/emoji/emoji-frequency/)¹, l'organizzazione no-profit responsabile della digitalizzazione delle lingue del mondo, le “lacrime di gioia” (😄) rappresentano oltre il 5% di tutte le emoji utilizzate (l'unico altro carattere che ci si avvicina è il ❤️). Le prime dieci emoji utilizzate in tutto il mondo sono: 😄❤️🎉👍🙌🙏😏😘😍😊. Quando scambiate messaggi su Telegram (o sulla vostra app di messaggistica preferita), quali sono le tre emoji che voi personalmente usate più di frequente? Utilizzando le informazioni sulla codifica Unicode [qui raccolte](https://unicode-table.com/)², scrivere un programma che mostri a schermo, per ciascuna di queste tre emoji:

- I. la posizione in classifica (rank);
- II. il Numero Unicode;
- III. il Nome Unicode;
- IV. l'emoji stessa.

Formattare l'output in modo che le informazioni relative a ciascuna delle tre emoji siano raccolte su una riga, e i diversi campi siano allineati verticalmente.

02.2.5 Immatricolazioni. La matricola degli studenti di un Ateneo è composta da due parti: una lettera e una sequenza di numeri. Scrivere un programma che, a partire da due codici di matricola, li mostri a schermo in ordine crescente in base alla sola parte numerica. Suggerimento: utilizzare le funzioni predefinite del linguaggio Python.

¹ <https://home.unicode.org/emoji/emoji-frequency/>

² <https://unicode-table.com/>