

## Esercitazione Python n. 6 -- 8 Novembre 2022

Obiettivo dell'esercitazione è prendere confidenza con Python e con l'ambiente IDLE.

In laboratorio, per avviare la macchina virtuale, dovete selezionarla all'interno della schermata di VirtualBox (schermata visualizzata all'accensione del PC), cliccando sull'opzione **LXLE-BIAR-4.3**

Una volta avviata la macchina virtuale, svolgete gli esercizi così come indicato nel testo. Scrivete i vostri programmi nei file che abbiamo predisposto: Esercizio 1 nel file `esercizio1.py`, Esercizio 2 nel file `esercizio2.py`, e così via. Per farlo usare l'ambiente IDLE di Python. Ovviamente è possibile consultare il materiale didattico disponibile sulla pagina web del corso (<https://classroom.google.com/u/0/c/NTQ1Njg4NzE1ODA5>). Si ricorda che le note relative alle lezioni Python possono essere lette con l'applicazione Jupyter Notebook. Sul sito del corso è comunque distribuita anche una versione pdf delle stesse.

**La consegna deve essere effettuata entro le 23:59 di Mercoledì 9 novembre.**

LE ESERCITAZIONI SVOLTE CONSEGNATE OLTRE QUESTO TERMINE, O CHE NON RISPETTANO IL FORMATO INDICATO PER LA CONSEGNA, NON VERRANNO CONSIDERATE.

In particolare, vi chiediamo di NON caricare un esercizio svolto per volta, di NON usare formati di compressione diversi da .zip, di NON rinominare i file o metterli in sottocartelle.

Fate attenzione che gli input siano richiesti all'utente UNO PER VOLTA e NELL'ORDINE RIPORTATO nell'esercizio e che le vostre stampe riportino a video i messaggi ESATTAMENTE nel formato atteso.

## Esercizi

1. Scrivere un programma Python che prende in input due numeri interi **n1** e **n2** maggiori di 0 e stampa in ordine decrescente tutti i divisori di **n1** che NON sono divisori di **n2**. Ad esempio, se **n1** vale 24 e **n2** vale 6 allora deve stampare (uno per riga): 24, 12, 8, 4.
2. Diciamo che una stringa è palindroma di livello 1 se il primo e ultimo carattere sono uguali, di livello 2 se il primo è uguale all'ultimo e il secondo al penultimo, e così via. Scrivere un programma Python che prende in input una stringa **s** e calcola il livello massimo di palindromicità della stringa. Ad esempio, se la stringa **s** vale 'alidfefcila' allora il livello è 3. Se la stringa è palindroma allora il suo livello deve essere uguale alla sua lunghezza.
3. Scrivere un programma Python che stampa la tabellina pitagorica dei numeri in base ottale. Usare due cifre, e.g.,  $02 \times 04 = 10$ .
4. Due viaggiatori partono insieme a velocità differenti: il primo viaggiatore viaggia alla velocità di 20 chilometri al giorno; il secondo viaggia alla velocità di 1 chilometro al giorno il primo giorno, 2 il secondo, 3 il terzo e così via (problema posto da Fibonacci nel 1200). Scrivere un programma Python, parametrico rispetto alle due velocità, che stampa il numero di giorni necessari al primo viaggiatore per raggiungere il secondo, senza usare la formula della successione. Usare un ciclo for con mille iterazioni con un **break** al suo interno.
5. Scrivere un programma Python che riceve in ingresso una stringa alfanumerica non vuota **s** e stampa il carattere con più occorrenze consecutive e il numero di occorrenze dello stesso. Se due caratteri condividono il massimo numero di occorrenze il programma stampa il secondo. Ad esempio, se **s** vale 'a32ppp7666' il programma stampa **6 3**.
6. Un ipotetico messaggio dallo spazio fa scoprire una razza extraterrestre umanoide (due gambe, due braccia). Dal messaggio si scopre che gli extraterrestri rappresentano i numeri con una notazione posizionale e nel messaggio è presente una equazione  $(\text{numeroTerrestre})_{10} = (\text{numeraleAlieno})_x$ . Scrivere un programma Python che stampa il numero di dita degli extraterrestri. N.B.: i valori `numeroTerrestre` e `numeraleAlieno` sono già presenti nel codice e si assume che gli alieni usino una base **x** uguale al numero delle loro dita.

7. Scrivere un programma Python che riceve in ingresso due stringhe **s1** e **s2** non vuote e cerca e stampa la sequenza di caratteri più lunga di **s1** presente in **s2**. Ad esempio, se **s1**='casaletto' e **s2**='salerno' il programma stampa **sale**. Usare un doppio ciclo for con un **break** nel ciclo interno.
8. Scrivere un programma Python che prende in input 2 stringhe **s1** ed **s2** e un numero intero **n** e calcola una nuova stringa composta dei caratteri di **s1** che sono presenti anche in **s2** in una posizione distante al massimo **n** posizioni (prima o dopo) da quella che hanno in **s1**. Ad esempio, se **s1** vale 'cestello', **s2** vale 'sportina' ed **n** vale 2, allora il risultato deve essere 'st'