

Esercitazione Python n. 4 -- 24 Ottobre 2023

Obiettivo dell'esercitazione è prendere confidenza con l'uso delle funzioni in Python, ripassare le istruzioni **for** e **if-then-else** ed esercitarsi con l'istruzione **while**.

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-4.5**. Una volta avviata la macchina virtuale, svolgete gli esercizi così come indicato nel testo. Usate l'ambiente Spyder per svolgere gli esercizi. Ovviamente è possibile consultare il materiale didattico disponibile sulla pagina web del corso (<https://classroom.google.com/u/1/c/NjIwOTY0ODk3MDAx>). Si ricorda che le note relative alle lezioni Python possono essere lette con l'applicazione JupyterLab.

La consegna deve essere effettuata entro l'orario di fine dell'esercitazione.

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.

Ogni esercizio richiede che sia completata una funzione all'interno del file predisposto con lo stesso nome e, come visto a lezione, può essere testata eseguendo il file stesso, con i casi di test forniti. Notate infatti che ogni file incorpora del codice python per eseguire alcuni test sulla funzione. **NON modificate questo codice, ma SCRIVETE SOLO il contenuto della funzione. Non spostate i file dalla loro posizione e non create nuovi file. Si noti che, come in sede di esame, per la correzione verranno usati insieme di dati di test diversi.**

ESERCIZI

- 1) **A_Ex1(s1, s2)** Scrivere una funzione che prende in ingresso due stringhe **s1** e **s2** e restituisce il prefisso più lungo comune alle due stringhe. Ad esempio, se **s1** è 'amaca' e **s2** è 'amaranto', la funzione deve restituire la stringa 'ama'. Se invece, **s1** è 'amaca' e **s2** è 'stringa' la funzione deve restituire la stringa vuota ''. Se (almeno) una delle due stringhe è vuota la funzione deve restituire ''.
- 2) **A_Ex2(s)** Scrivere una funzione che prende in ingresso una stringa **s** composta da almeno un carattere e la scandisce carattere dopo carattere partendo dal primo, sino a quando o la stringa finisce o si incontra un carattere il cui codice Unicode è maggiore di 100. La funzione restituisce:
 - La stringa vuota se la stringa è stata consumata senza trovare il carattere.
 - Il primo carattere con codice Unicode maggiore di 100, nel caso in cui si trova un carattere il cui codice Unicode è maggiore di 100

Esempio:

- Se **s** è "CONTE", restituisce ""

- Se **s** è “abaco”, restituisce “o”
- Se **s** è “adamo”, restituisce “m”

3) **A_Ex3(n)** Scrivere una funzione che prende in ingresso un intero **n** e restituisce `True` se **n** è un numero primo, `False`, altrimenti.

Esempio:

- Se **n** è “7”, restituisce `True`
- Se **n** è “4096”, restituisce `False`

4) **A_Ex4(n1, n2)** Scrivere una funzione che prende in ingresso due numeri interi **n1** e **n2** e restituisce il loro prodotto calcolato senza usare l'operatore * (moltiplicazione) o / (divisione).

Esempio:

- Se **n1** e **n2** sono “5” e “2”, restituisce “10”
- Se **n1** e **n2** sono “-3” e “4”, restituisce “-12”

Nota: Gli interi in input possono essere positivi, negativi o pari a 0.

5) **A_Ex5(s)** Scrivere una funzione che prende in ingresso una stringa **s** e restituisce il carattere che compare più volte in **s**. Se c'è più di un carattere con queste caratteristiche, restituisce quello che fra questi ha tra le sue occorrenze l'indice maggiore.

Esempio:

- Se **s** è “pippo”, restituisce “p”
- Se **s** è “clarabella”, restituisce “a” (sia ‘a’ che ‘l’ compaiono 3 volte, che è il massimo numero di volte in cui compare un carattere nella stringa, ma c'è una occorrenza di ‘a’ che ha l'indice maggiore di tutte le occorrenze di ‘l’)

Nota: Si assuma che la stringa **s** in input non sia mai la stringa vuota.

6) **A_Ex6(s1, s2)** Scrivere una funzione che prende in ingresso due stringhe **s1** e **s2** non nulle, della stessa lunghezza e corrispondenti a due numerali da interpretare CP2 e restituisce una stringa corrispondente al numerale in CP2 della somma di **s1** e **s2** (se la somma ha avuto successo) o 'ERRORE' in tutti gli altri casi. Si ricorda che in una somma in CP2 $0+1=1$, $1+1=0$ col riporto di 1 e $1+1+1=1$ col riporto di 1. Inoltre, si ricorda che se gli addendi hanno segni differenti il risultato è sempre corretto; se hanno lo stesso segno il risultato è corretto solo se ha lo stesso segno degli addendi.

Esempio:

- Se **s1** è “000” e **s2** è “01” restituisce “ERRORE”
- Se **s1** è “101” e **s2** è “001” restituisce “110”

7) **A_Ex7(ex, cp2)** Scrivere una funzione che prende in ingresso, nell'ordine, due stringhe **ex** e **cp2** non nulle, della stessa lunghezza e corrispondenti a due numerali da interpretare in eccesso e CP2, rispettivamente; la funzione restituisce 'ERRORE' se la lunghezza delle stringhe non è corretta, 'ex' se $(ex)_{ex} \geq (cp2)_{cp2}$, altrimenti.

Esempio:

- Se **ex** è “000” e **cp2** è “01” restituisce “ERRORE”
- Se **ex** è “001” e **cp2** è “001” restituisce “cp2”

- 8) **A_Ex8(numBin)** Scrivere una funzione che prende in ingresso una stringa **numBin** corrispondente a un binario puro con parte frazionaria nella forma di una sequenza di caratteri '01' con un punto al suo interno (almeno un carattere a sinistra del punto, almeno un carattere a destra del punto), e.g., '101.01'. La funzione assume sempre corretto il formato della stringa e restituisce una stringa corrispondente al valore in base 10 di **numBin**.

Esempio:

- Se **numBin** è "1.1" la funzione restituisce "1.5"
- Se **numBin** è "0.01" la funzione restituisce "0.25"

- 9) Scrivere una funzione che riceve in ingresso una stringa **s** composta da zeri e uno e un intero **n** $\geq \text{len}(s)$ e calcola il CP2 di **s** esteso a **n** bit, ovvero ricopia sino al primo 1 compreso, poi complementa ed estende il numerale a **n** bit. Assumere che l'ingresso sia corretto.

Esempio:

- Se **s** vale '11' e **n** = 5 la funzione deve restituire '00001'.

- 10) Scrivere una funzione che riceve in ingresso una stringa **s** e due caratteri **c1** e **c2** non necessariamente distinti e non necessariamente presenti in **s**; la funzione calcola la lunghezza della più corta sequenza di caratteri consecutivi di **s** che contiene sia **c1** che **c2**.

Esempio:

- Se **s** vale 'casale' e **c1** = 'l' e **c2** = 'a', la funzione deve restituire 2;
- Se **s** vale 'casale' e **c1** = 'a' e **c2** = 'a', la funzione deve restituire 1;
- Se **s** vale 'casale' e **c1** = 'z' e **c2** = 'f', la funzione deve restituire 0.