

Esercitazione Python n. 4 -- 26 Ottobre 2021

Obiettivo dell'esercitazione è scrivere i primi programmi che fanno uso dell'istruzione while e delle variabili accumulatori. Per risolvere gli esercizi descritti di seguito dovete modificare i file "esercizio1.py", "esercizio2.py", ecc., che trovate nella cartella dell'esercitazione.

Per **consegnare l'esercitazione svolta**, comprimate nuovamente la cartella LabPython04 in un file .zip e caricatela sulla pagina del corso dal link <https://classroom.google.com/u/0/w/MTY5MzU2NjY1MTg2/t/all> (Dalla sezione 'Esercitazioni su Python', selezionate 'Esercitazione 4' e successivamente 'Visualizza Compito'; poi cliccate su 'Aggiungi o crea' e scegliete il file da caricare). Al termine dell'operazione cliccate su 'Contrassegna come completato'. Vi chiediamo di effettuare la consegna entro le 23:59 di Mercoledì 27 ottobre.

Si rammenta che le esercitazioni che non rispettano le indicazioni per la consegna non possono essere considerate per l'assegnazione del punto bonus. In particolare, ricordate di comprimere e riconsegnare l'intera cartella dell'esercitazione, e non singolarmente i file degli esercizi, di NON usare formati di compressione diversi da .zip, di NON rinominare i file o metterli in sottocartelle. Inoltre, prestate attenzione ad input ed output degli esercizi: gli **input** devono essere richiesti all'utente **UNO PER VOLTA** e **NELL'ORDINE INDICATO** nel testo dell'esercizio; l'**output** restituito (quando richiesto) deve essere **STAMPATO ESATTAMENTE** nel formato indicato dall'esercizio, che potete verificare anche facendo riferimento agli esempi riportati (ad esempio, nell'esercizio 1, se vengono inseriti "pippo" "albero" e "casa" il vostro programma deve stampare a schermo "3", e non "il risultato è 3" o altri messaggi del genere).

In ogni esercizio, se non diversamente richiesto, **potete sempre assumere che gli input forniti siano coerenti con la traccia** (cioè, se l'esercizio chiede di dare in input al programma un numero intero, potete assumere che l'input sia sempre convertibile in intero, e non è necessario nel codice effettuare controlli in tal senso).

Esercizi

- 1) Scrivere un programma python che chiede in input all'utente una sequenza di stringhe, una per una in maniera iterativa, terminando la richiesta di inserimento in input quando viene immessa una stringa contenente sia il carattere 'a' che il carattere 'c', e stampa il numero di stringhe lette (inclusa l'ultima).

Esempio:

- Inserendo in questo ordine le stringhe "pippo", "albero", "casa" il programma termina stampando "3"

- 2) Scrivere un programma python che chiede in input all'utente una sequenza di numeri interi, uno per uno in maniera iterativa terminando la richiesta di inserimento in input quando viene immesso il carattere "*" (asterisco), e stampa a schermo quanti interi positivi sono stati inseriti.

Esempio:

- Inserendo gli interi "4", "-5", "-6", "3" e infine il carattere "*", il programma stampa "2"
- Inserendo gli interi "-5" "-5" e infine il carattere "*", il programma stampa "0"

- 3) Scrivere un programma python che chiede in input all'utente una sequenza di numeri interi, uno per uno in maniera iterativa terminando la richiesta di inserimento in input quando viene immesso il carattere "*" (asterisco), e stampa a schermo la somma dei soli interi negativi.

Esempio:

- Inserendo gli interi "14", "-4", "-7", "22" e infine il carattere "*", il programma stampa "-11"
- Inserendo gli interi "5" "-5" e infine il carattere "*", il programma stampa "-5"

- 4) Scrivere un programma python che chiede in input all'utente una sequenza di numeri interi, uno per uno in maniera iterativa, terminando la richiesta di inserimento in input quando viene immesso l'intero "0", e stampa a schermo il risultato della somma dei soli interi divisibili per 3.

Esempio:

Inserendo in questo ordine gli interi "1", "2", "-6", "9", "0" il programma termina e stampa "3"

- 5) Scrivere un programma python che chiede in input all'utente un intero maggiore o uguale a 0 e stampa il suo fattoriale. Si ricorda che il fattoriale un numero intero n , indicato con $n!$, è pari a 1 se $n=0$ e se $n=1$, ed è invece pari al prodotto dei numeri interi positivi minori o uguali a n se $n>1$.

Esempio:

- Inserendo l'intero "5", il programma stampa "120" ottenuto in quanto $5! = 5*4*3*2*1$

- 6) Scrivere un programma python che chiede in input all'utente due numeri interi e ne stampa il loro prodotto calcolato senza usare l'operatore * (moltiplicazione) o / (divisione).

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

- 7) Scrivere un programma python che chiede in input all'utente una stringa s e stampa a schermo il carattere che compare più volte in s (distinguendo fra maiuscole e minuscole). Se c'è più di un carattere con queste caratteristiche, stampare quello che fra questi compare per ultimo nella stringa (da sinistra a destra).

Esempio:

- Inserendo la stringa "pippo", il programma deve stampare "p"
- Inserendo la stringa "clarabella", il programma deve stampare "a" (sia 'a' che 'l' compaiono 3 volte, che è il massimo numero di volte in cui compare un carattere nella stringa, e c'è una occorrenza di 'a' che segue tutte le occorrenze di 'l', per cui 'a' compare per ultimo)

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

- 8) Scrivere un programma che prende in ingresso una sequenza di stringhe, una per una in maniera iterativa, e termina la richiesta di inserimento in input quando l'ultimo carattere della stringa precedente è uguale al primo carattere di quella attuale. Prima di terminare il programma stampa su un'unica riga le ultime due stringhe inserite, separate da uno spazio.

Esempio:

- Inserendo in questo ordine le stringhe "pippo", "casa", "albero", il programma termina stampando "casa albero".

Nota: Si assuma che il programma riceva in ingresso sempre almeno due stringhe e che le stringhe in ingresso non siano mai vuote.

- 9) Scrivere un programma python che chiede in input all'utente un intero n maggiore di 0 e stampa uno per riga i primi n numeri della successione di Fibonacci. I primi due numeri della successione di Fibonacci sono 1 ed 1, ed ogni numero successivo è dato dalla somma dei due precedenti. Quindi avremo: 1 1 2 3 5 8 13 21 34...

Esempio:

Inserendo l'intero "6", il programma stampa "1", a capo "1", a capo "2", a capo "3", a capo "5", a capo "8".

- 10) Scrivere un programma python che chiede in input all'utente una sequenza di stringhe, una per una in maniera iterativa terminando la richiesta di inserimento quando la somma delle lunghezze di due stringhe adiacenti (cioè immesse una dopo l'altra) è uguale alla lunghezza della stringa successiva. Prima di terminare il programma stampa su un'unica riga le tre stringhe coinvolte nel confronto, separate da uno spazio.

Esempio:

- Inserendo in questo ordine le stringhe "fondamenti", "uno", "due", "FI1", "darwin" il programma termina stampando "due FI1 darwin".

Nota: Si assuma che il programma riceva sempre in ingresso almeno tre stringhe.