

## INDICE

INDICE.....	1
INTRODUZIONE ED OBIETTIVO.....	2
DESCRIZIONE E SVOLGIMENTO.....	3
VERIFICA RISULTATO FINALE.....	4
CONCLUSIONE.....	5
GLOSSARIO.....	6

## INTRODUZIONE ED OBIETTIVO

**INTRODUZIONE:** Attraverso la creazione di un piccolo programma, si intende comprendere come iterare sugli elementi di una lista e come ottenere nuovi valori derivati da quelli originali, applicando operazioni base su stringhe e numeri.

Per rendere l'attività più piacevole, è stato scelto un contesto a tema culinario ispirato , con una punta di ironia , a quel leggero languorino che spesso accompagna le sessioni di studio serali. Questa scelta permette di unire la logica della programmazione ad un esempio concreto e immediatamente comprensibile, mantenendo al tempo stesso un tono leggero ma professionale.

**OBIETTIVO:** L'obiettivo dell'esercizio è progettare una funzione capace di ricevere in ingresso una lista di parole e di restituire in uscita una nuova lista composta da valori numerici corrispondenti alla lunghezza di ciascuna parola.

Il codice, tramite un ciclo e l'utilizzo del metodo **append()**, elabora i dati in modo strutturato evidenziando il processo logico di trasformazione da input testuale a output numerico.

## DESCRIZIONE E SVOLGIMENTO

```
1 #ESERCIZIO M2w7D1
2
3 def lunghezze_parole(A): #DEFINIZIONE FUNZIONE
4     B = [] #LISTA VUOTA, VERRANNO MEMORIZZATE LE LUNGHEZZE DELLE PAROLE
5     for parola in A: #SCORRE TUTTI GLI ELEMENTI DELLA LISTA A, UNO ALLA VOLTA, ASSEGNANDO OGNI PAROLA ALLA VARIABILE 'PAROLA'
6         B.append(len(parola)) #CALCOLA LA LUNGHEZZA DELLA PAROLA E LA AGGIUNGE(APPEND) ALLA LISTA B
7     return B #RESTITUISCE LA LISTA B COME RISULTATO DELLA FUNZIONE
8
9
10 A = ["pizza", "lasagna", "tiramisu", "risotto", "gelato"] #CREO LISTA A CONTENENTE ALCUNE PAROLE
11 B = lunghezze_parole(A) #CHIAMO LA FUNZIONE
12 print(B) #STAMPO A SCHERMO LA LISTA B OSSIA LE LUNGHEZZE DELLEAPROLE CONTENUTE IN A
13
14
```

Per lo sviluppo dell'esercizio è stata definita una **funzione** chiamata **lunghezze\_parole**, progettata per ricevere in ingresso una lista di parole e restituire una nuova lista contenente la lunghezza di ciascun elemento.

All'interno della funzione viene creata una **lista vuota** denominata **B**, che rappresenta la struttura dati destinata a raccogliere i risultati.

Successivamente attraverso un **ciclo for** il programma scorre ogni parola della lista di partenza A, calcola la sua lunghezza tramite la **funzione integrata len()** e la aggiunge progressivamente alla lista B mediante il metodo **append()**.

Una volta terminato il ciclo, la funzione restituisce la lista completa, che viene poi visualizzata tramite l'**istruzione print()**.

Per rendere l'attività più piacevole, è stata scelta una lista di parole legata al mondo della cucina **"pizza", "lasagna", "tiramisu", "risotto" e "gelato"** un piccolo richiamo simbolico al tema culinario che accompagna l'esercizio, come se ogni parola fosse un ingrediente del programma. L'esecuzione del codice ha permesso di ottenere in output una nuova lista di numeri interi, ciascuno corrispondente alla lunghezza delle parole inserite nella lista originale, dimostrando così il corretto funzionamento della logica implementata.

## VERIFICA RISULTATO FINALE

```
>>  
[5, 7, 8, 7, 6]  
PS C:\EP. ESERCIZI\PYTHON> 
```

L'esecuzione del programma ha prodotto come **risultato la lista [5, 7, 8, 7, 6]** valore che rappresenta la lunghezza delle parole contenute nella lista di partenza.

Ogni numero corrisponde precisamente al conteggio dei caratteri di ciascun termine: "pizza" è composta da 5 lettere, "lasagna" da 7, "tiramisu" da 8, "risotto" da 7 e "gelato" da 6.

Il risultato conferma che la funzione ha operato correttamente, traducendo una sequenza di parole in una nuova forma numerica coerente e perfettamente aderente alla logica richiesta dalla traccia.

## CONCLUSIONE

L'esercizio ha permesso di consolidare l'utilizzo delle funzioni e delle strutture iterative, mostrando come anche operazioni apparentemente semplici (come il conteggio dei caratteri di una parola ) possano essere gestite in modo strutturato ed elegante attraverso il codice.

La funzione realizzata rappresenta un esempio chiaro di come consenta di trasformare dati testuali in valori numerici, applicando logica e metodo a ogni fase del processo.

Dal punto di vista formativo, l'attività ha evidenziato l'importanza della chiarezza nel flusso di esecuzione e dell'organizzazione dei dati.

Il contesto culinario scelto, oltre ad alleggerire l'approccio, ha reso l'esperienza più "leggera" dimostrando che anche nella programmazione proprio come in cucina precisione, ordine e un pizzico di creatività possono trasformare pochi ingredienti in un risultato armonioso e ben riuscito.

## GLOSSARIO

**Funzione (def)** – Struttura che consente di racchiudere un insieme di istruzioni all'interno di un blocco riutilizzabile.

In questo esercizio la **funzione lunghezze\_parole()** è stata utilizzata per eseguire un'operazione specifica su una lista di parole.

**Lista (list)** – Struttura dati che consente di memorizzare una sequenza ordinata di elementi. Può contenere parole, numeri o altri tipi di dati e permette di accedere a ciascun elemento tramite un indice.

**Ciclo for** – Istruzione iterativa che consente di ripetere un blocco di codice per ogni elemento presente in una sequenza (come una lista o una stringa). È utilizzato per scorrere gli elementi della lista di parole e calcolarne la lunghezza.

**Metodo append()** – Metodo associato alle liste che consente di aggiungere un nuovo elemento alla fine della lista stessa.

È impiegato per inserire progressivamente le lunghezze delle parole nella lista B.

**Funzione len()** – Funzione che restituisce la lunghezza (cioè il numero di caratteri) di una stringa o il numero di elementi di una sequenza.

**Istruzione print()** – Comando utilizzato per mostrare a schermo il risultato di un'elaborazione, utile per verificare il corretto funzionamento del programma.

**Variabile** – Spazio di memoria identificato da un nome, usato per conservare dati che possono cambiare durante l'esecuzione del programma.

**Return** – Parola chiave che consente di restituire un valore al termine dell'esecuzione di una funzione. Nel programma permette di passare la lista B come risultato dell'elaborazione.