Lezione S2/L3

Linguaggio python

La Lezione di oggi ha richiesto lo sviluppo di un programma utilizzando il linguaggio di programmazione python. in questo programma dovevamo:

- Scrivere un programma in Python che genera un nome per una band musicale utilizzando due input forniti dall'utente: la città di origine e il nome del proprio animale domestico.

```
#!/usr/bin/env python3

def genera_nome_band():
    print("Benvenuto nel generatore di nomi per band!")

    # Chiedi all'utente la città di origine
    citta = input("Inserisci la città di origine: ").strip()

    # Chiedi all'utente il nome del proprio animale domestico
    animale = input("Inserisci il nome del tuo animale domestico: ").strip()

    # Genera il nome della band combinando i due input
    nome_band = f"{citta} {animale}"

    # Mostra il nome della band
    print(f"Il nome della tua band potrebbe essere: {nome_band}")


# Esegui la funzione
genera_nome_band()
```

figura 1: Il codice

Analizziamo ora le varie parti del codice e capiamo come siamo arrivati a questo risultato. Personalmente ho svolto il programma su terminale mac, i comandi sono praticamente uguali e il procedimento è esattamente uguale.
Innanzitutto dobbiamo creare il file ”NomeBand.py” per fare questo apriamo il terminale, viaggiamo nella directory dove vogliamo lavorare ed utilizziamo il comando nano come nell’esercizio precedente.

il codice si divide in 3 parti (figura 2):

INIZIALIZZAZIONE : per qualche ragione se provo ad eseguire tramite il comando ./NomeBand.py il programma non viene eseguito con un interprete Python, devo quindi specificare come fare aggiungendo questa linea di codice chiamata **shebang.**

FUNZIONE : per rendere l’esercizio ordinato e semplice ho deciso di creare un’unica funzione che svolge tutti i compiti elencati dall’esercizio. il comando .**strip()** viene utilizzato per rimuovere eventuali spazi bianchi (o caratteri specifici) all'inizio e alla fine di una stringa.

Inoltre , il prefisso **f** davanti a una stringa indica che quella stringa è una **stringa formattata**, o **f-string**. È un modo comodo e leggibile per incorporare variabili o espressioni direttamente all'interno di una stringa.

CALCOLI: l'ultima parte chiama la funzione che quindi, quando si svolgerà il programma, effettuerà i passaggi specificati nella funzione.

```
#!/usr/bin/env python3

def genera_nome_band():
    print("Benvenuto nel generatore di nomi per band!")

    # Chiedi all'utente la città di origine
    citta = input("Inserisci la città di origine: ").strip()

    # Chiedi all'utente il nome del proprio animale domestico
    animale = input("Inserisci il nome del tuo animale domestico: ").strip()

    # Genera il nome della band combinando i due input
    nome_band = f"{citta} {animale}"

    # Mostra il nome della band
    print(f"Il nome della tua band potrebbe essere: {nome_band}")

# Esegui la funzione
genera_nome_band()
```

figura 2: breakdown del codice

Dopo aver salvato il programma con "**ctrl +O**" possiamo uscire e senza necessità di creare un eseguibile digitiamo, **./NomeBand.py**.
nel mio caso la CLI mi restituisce un errore dicendomi che non ho i permessi per eseguire questo programma. questo viene fatto per ragioni di sicurezza in automatico per evitare che un utente esterno possa eseguire programmi non autorizzati.
ciò che dobbiamo fare è quindi modificare i diritti di esecuzione utilizzando il comando:

chmod +rwx NomeBand.py

specificatamente modifichiamo i diritti in lettura(r), scrittura(w) ed esecuzione(x) per l'utente principale del file NomeBand.py (figura 3). Verifichiamo quindi i diritti tramite il comando:
ls -la.

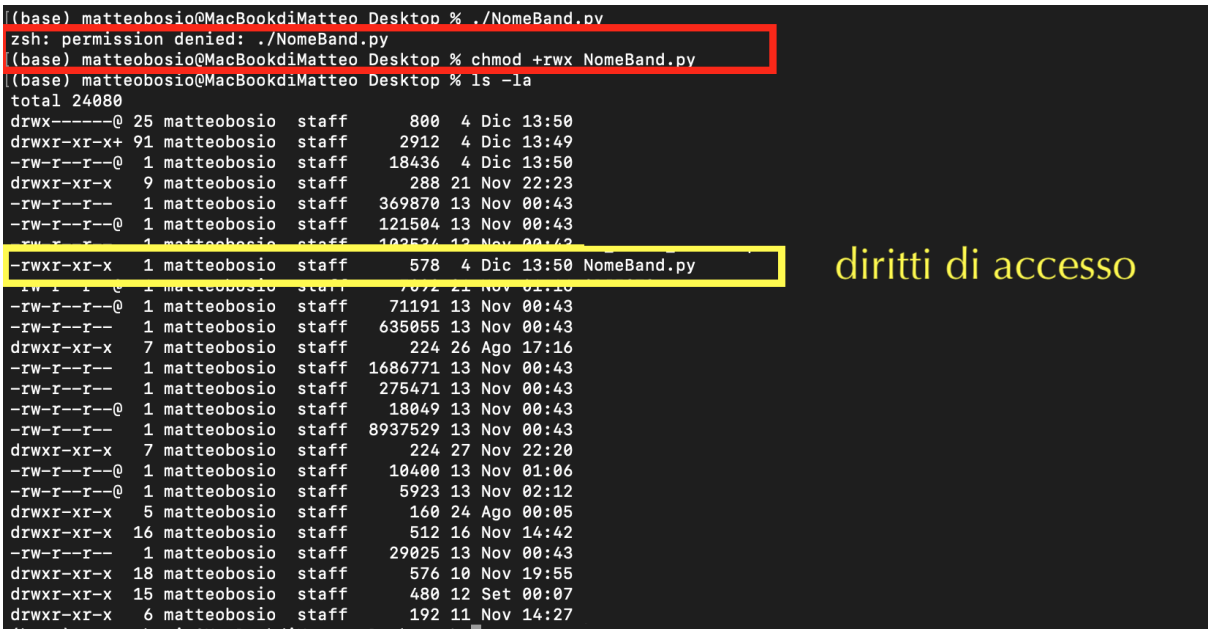


figura 3: Diritti Programma

Possiamo adesso utilizzare il programma e verificare il corretto funzionamento:

figura 4: Esecuzione del programma

BONUS 1

in questo esercizio dovevamo Scrivere una funzione che calcoli la media mobile di una lista di numeri. Personalmente ho aggiunto:

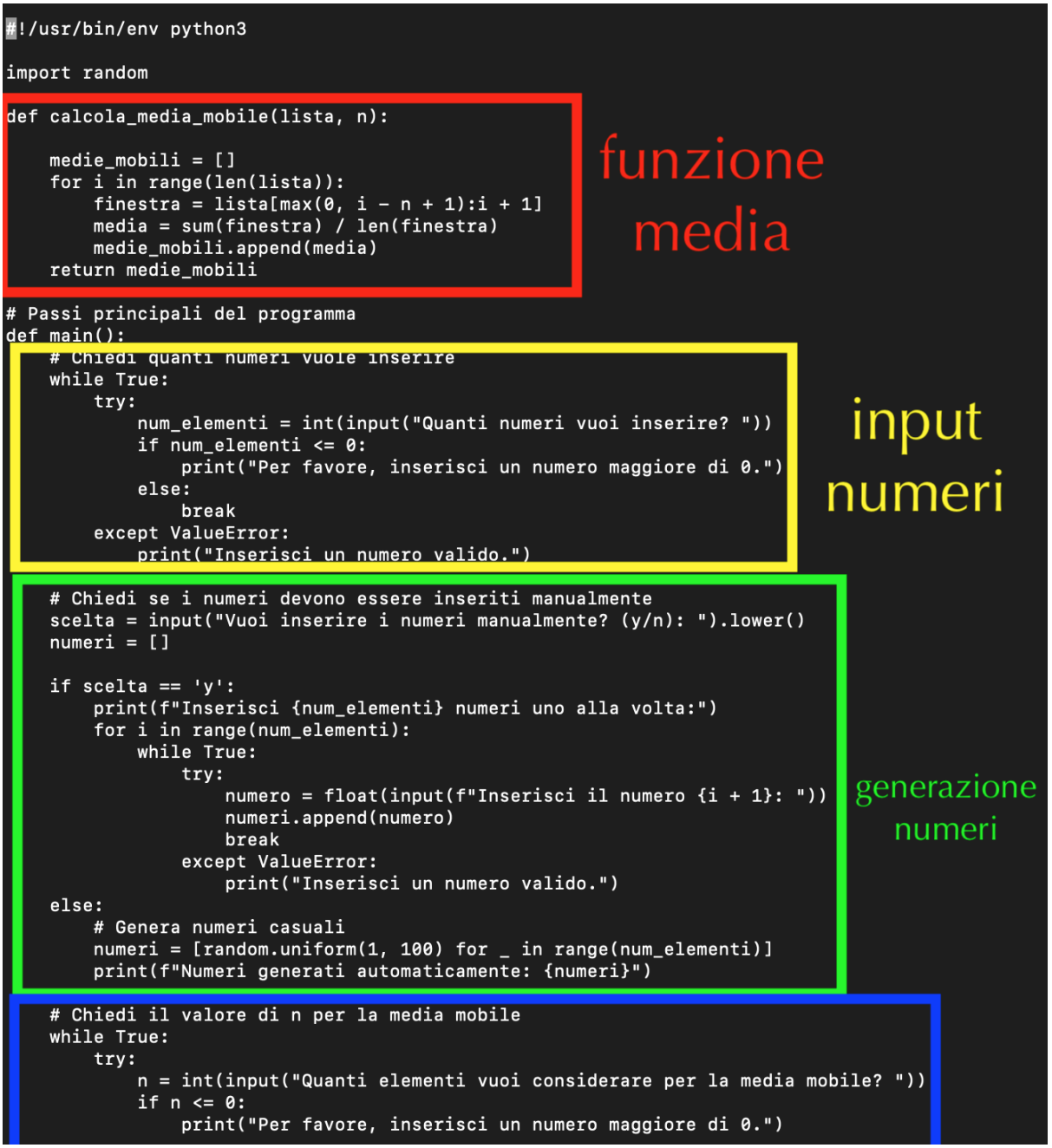
- quanti valori si vogliono inserire.
- se si vogliono inserire manualmente o automaticamente.
- e quanti elementi si vogliono prendere per la media.

```
#!/usr/bin/env python3

import random

def calcola_media_mobile(lista, n):

    medie_mobili = []
    for i in range(len(lista)):
        finestra = lista[max(0, i - n + 1):i + 1]
        media = sum(finestra) / len(finestra)
        medie_mobili.append(media)
    return medie_mobili

# Passi principali del programma
def main():
    # Chiedi quanti numeri vuole inserire
    while True:
        try:
            num_elementi = int(input("Quanti numeri vuoi inserire? "))
            if num_elementi <= 0:
                print("Per favore, inserisci un numero maggiore di 0.")
            else:
                break
        except ValueError:
            print("Inserisci un numero valido.")

    # Chiedi se i numeri devono essere inseriti manualmente
    scelta = input("Vuoi inserire i numeri manualmente? (y/n): ").lower()
    numeri = []

    if scelta == 'y':
        print(f"Inserisci {num_elementi} numeri uno alla volta:")
        for i in range(num_elementi):
            while True:
                try:
                    numero = float(input(f"Inserisci il numero {i + 1}: "))
                    numeri.append(numero)
                    break
                except ValueError:
                    print("Inserisci un numero valido.")
    else:
        # Genera numeri casuali
        numeri = [random.uniform(1, 100) for _ in range(num_elementi)]
        print(f"Numeri generati automaticamente: {numeri}")

    # Chiedi il valore di n per la media mobile
    while True:
        try:
            n = int(input("Quanti elementi vuoi considerare per la media mobile? "))
            if n <= 0:
                print("Per favore, inserisci un numero maggiore di 0.")
```

```
# Chiedi se i numeri devono essere inseriti manualmente
scelta = input("Vuoi inserire i numeri manualmente? (y/n): ").lower()
numeri = []

if scelta == 'y':
    print(f"Inserisci {num_elementi} numeri uno alla volta:")
    for i in range(num_elementi):
        while True:
            try:
                numero = float(input(f"Inserisci il numero {i + 1}: "))
                numeri.append(numero)
                break
            except ValueError:
                print("Inserisci un numero valido.")
else:
    # Genera numeri casuali
    numeri = [random.uniform(1, 100) for _ in range(num_elementi)]
    print(f"Numeri generati automaticamente: {numeri}")

# Chiedi il valore di n per la media mobile
while True:
    try:
        n = int(input("Quanti elementi vuoi considerare per la media mobile? "))
        if n <= 0:
            print("Per favore, inserisci un numero maggiore di 0.")
        else:
            break
    except ValueError:
        print("Inserisci un numero valido.")

# Calcola e mostra i risultati
medie_mobili = calcola_media_mobile(numeri, n)
print(f"\nArray completo: {numeri}")
print(f"Media mobile dinamica: {medie_mobili}")
```

figura 5: programma bonus 1

Vediamo le varie parti del programma :




```
#!/usr/bin/env python3

import random

def calcola_media_mobile(lista, n):

    medie_mobili = []
    for i in range(len(lista)):
        finestra = lista[max(0, i - n + 1):i + 1]
        media = sum(finestra) / len(finestra)
        medie_mobili.append(media)
    return medie_mobili

# Passi principali del programma
def main():
    # Chiedi quanti numeri vuole inserire
    while True:
        try:
            num_elementi = int(input("Quanti numeri vuoi inserire? "))
            if num_elementi <= 0:
                print("Per favore, inserisci un numero maggiore di 0.")
            else:
                break
        except ValueError:
            print("Inserisci un numero valido.")

    # Chiedi se i numeri devono essere inseriti manualmente
    scelta = input("Vuoi inserire i numeri manualmente? (y/n): ").lower()
    numeri = []

    if scelta == 'y':
        print(f"Inserisci {num_elementi} numeri uno alla volta:")
        for i in range(num_elementi):
            while True:
                try:
                    numero = float(input(f"Inserisci il numero {i + 1}: "))
                    numeri.append(numero)
                    break
                except ValueError:
                    print("Inserisci un numero valido.")
    else:
        # Genera numeri casuali
        numeri = [random.uniform(1, 100) for _ in range(num_elementi)]
        print(f"Numeri generati automaticamente: {numeri}")

    # Chiedi il valore di n per la media mobile
    while True:
        try:
            n = int(input("Quanti elementi vuoi considerare per la media mobile? "))
            if n <= 0:
                print("Per favore, inserisci un numero maggiore di 0.")
```

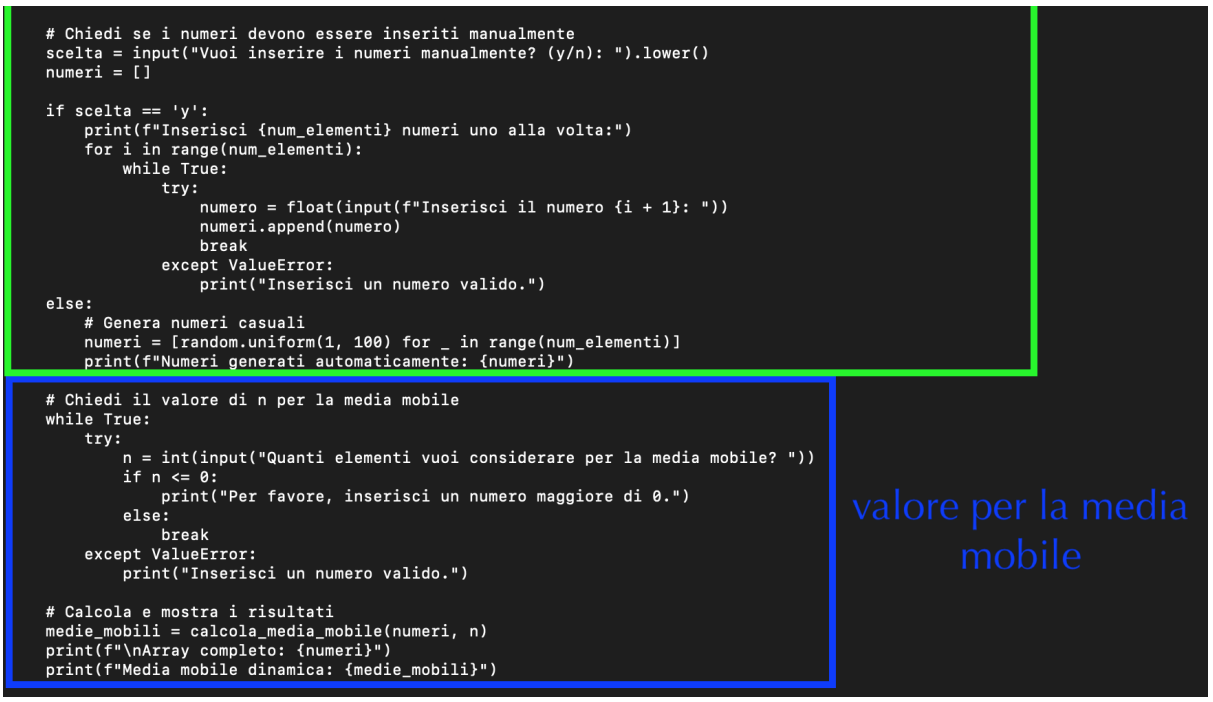


figura 6: breakdown

FUNZIONE MEDIA: la funzione prende in input una lista e un valore n e restituisce la media, ho utilizzato un ciclo for che calcola dinamicamente la media e la posiziona in medie mobili.

INPUT NUMERI: chiede quanti numeri inserire in totale ed effettua un controllo che il numero inserito sia più grande di zero e che continua a chiedere l'input in caso di errore (simile all'esercizio precedente).

GENERAZIONE NUMERI: qui il codice, tramite una condizione if chiede se si vuole inserire manualmente o automaticamente i numeri, in caso di scelta y allora il programma chiede l'inserimento di numeri che vengono poi allocati dinamicamente altrimenti, se si sceglie n, il programma genera automaticamente in maniera casuale dei numeri float compresi tra 1 e 100.

VALORE PER LA MEDIA MOBILE: qui viene chiesto quanti valori si usano di volta in volta per fare la media mobile con un check di sicurezza, e viene infinite calcolata la media tramite chiamata della funzione e vengono stampati sia i numeri scelti all'inizio che il vettore media mobile.

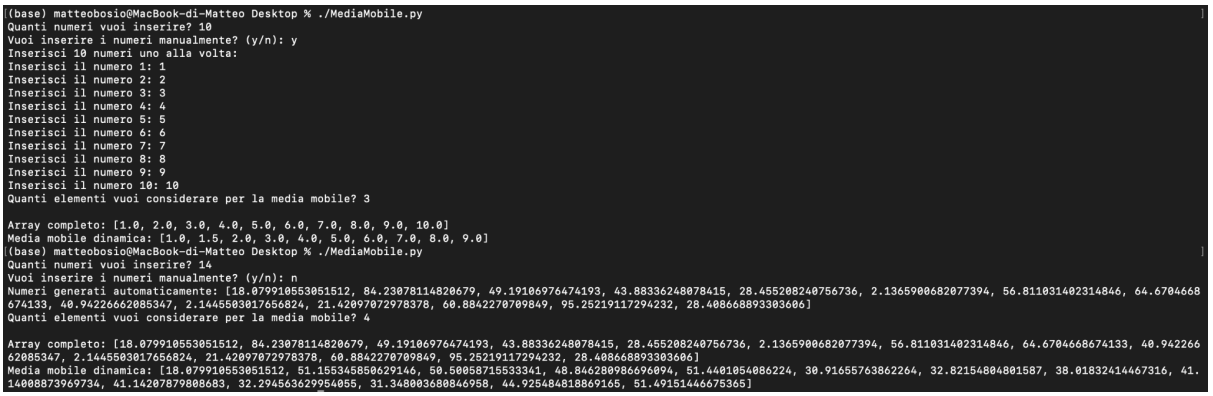


figura 7 : funzionamento del codice

BONUS 2

in questo esercizio dovevamo scrivere una funzione che analizzi una stringa di testo e restituisca un dizionario con il conteggio delle occorrenze di ciascuna parola. Ignorando la punteggiatura e considerando le parole in modo case-insensitive.

```
#!/usr/bin/env python3

import re

def conta_occorrenze(test):
    # Converti il testo in minuscolo
    testo_lower = test.lower()

    # Rimuovi la punteggiatura usando una regular expression
    testo_senza_punteggiatura = re.sub(r'[^\w\s]', '', testo_lower)

    # Splitta il testo in parole (usando gli spazi come delimitatori)
    parole = testo_senza_punteggiatura.split()

    # Crea un dizionario per contare le occorrenze delle parole
    dizionario_occorrenze = {}

    for parola in parole:
        # Incrementa il conteggio della parola nel dizionario
        if parola in dizionario_occorrenze:
            dizionario_occorrenze[parola] += 1
        else:
            dizionario_occorrenze[parola] = 1

    return dizionario_occorrenze

# Chiedi il testo all'utente
testo_utente = input("Inserisci il testo da analizzare: ")

# Calcola e stampa il risultato
risultato = conta_occorrenze(testo_utente)
print("Conteggio delle occorrenze delle parole:")
for parola, conteggio in risultato.items():
    print(f"{parola}: {conteggio}")
```

vediamo le varie parti:

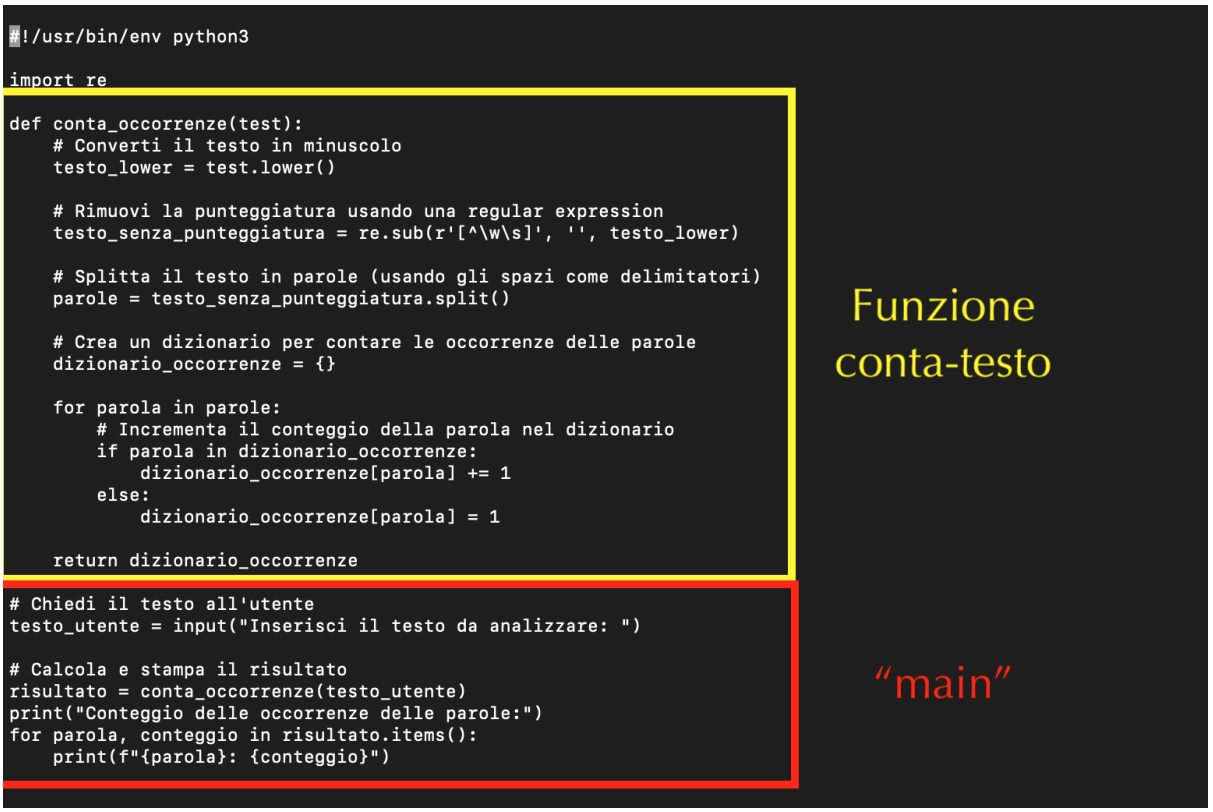


figura 8 : codice bonus 2

FUNZIONE CONTA-TESTO: Questa funzione converte prima tutto il testo in minuscolo per evitare problemi con i case sensitive, rimuove poi con l'utilizzo di una funzione nella libreria re la punteggiatura e tramite un ciclo for registra parole nuove oppure aggiunge un valore ad un contatore se sono uguali, ritorna alla fine queste occorrenze

MAIN: prima viene richiesto in input il testo, viene poi chiamata la funzione e viene poi fatto un ciclo for che stampa tutte le parole e il loro rispettivo conteggio.

questo è il risutlato:

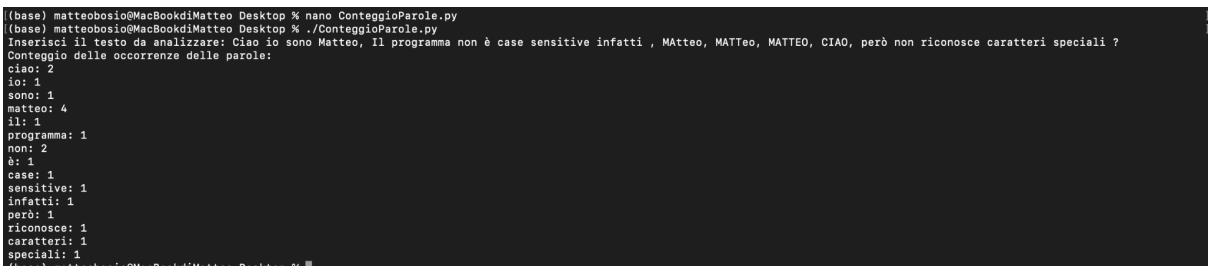


figura 9 : funzionamento codice bonus 2