

Come primo passaggio ho fatto inserire dall'utente le informazioni richieste dalla consegna:

```
x=input("inserisci il nome del tuo animale domestico: ")
y=input("inserisci la tua città: ")
print("il nome della tua band potrebbe essere: ", y+" "+x)
```

E le ho inserite rispettivamente nella variabile x, il nome dell'animale domestico, e nella variabile y, il nome della città.

Successivamente veniva chiesto di combinare il nome della città con quello dell'animale domestico e quindi ho usato un semplice "print" per stampare il risultato e con il simbolo "+" ho fatto la combinazione tra la variabile x e la variabile y.

```
(kali@kali)-[~/Documents/Epicode]
$ python primProvaPython.py
inserisci il nome del tuo animale domestico: Max
inserisci la tua città: Roma
il nome della tua band potrebbe essere: Roma Max
```

Questo è l'output

### Esercizi Bonus

Numero 1.

Per svolgere il primo esercizio bonus, ho inizialmente creato una lista di numeri di 10 elementi (10 a mia scelta), ho dichiarato e inizializzato la variabile n con il numero 3.

```
GNU nano 8.1 esercizio2S2.py
import os
os.system('clear')

lista=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
n=3
reversedLista=lista[::-1]

print("la lista: ", lista, "e' lunga: ", len(lista))
print("la lista al contrario e' ", reversedLista)
print("n= ", n)

for elemento in range(0, len(reversedLista)):
    finestra=reversedLista[elemento:n+elemento]
    somma=sum(finestra)
    media=somma/len(finestra)
    print(finestra, somma, "media: ", media)
    x=input("**")
```

La mia strategia era quella di girare al contrario la lista, così da avere l'ultimo elemento al posto del primo e viceversa per tutti gli elementi.

Poi ho mandato in stampa i dati che avevo inizializzato.

```
GNU nano 8.1      [esercizio252.py]
import os
os.system('clear')

lista=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
n=3
reversedLista=lista[::-1]

print ("la lista: ", lista, "e' lunga: ", len(lista))
print ("la lista al contrario e' ", reversedLista)
print ("n= ", n)

for elemento in range(0, len(reversedLista)):
    finestra=reversedLista[elemento:n+elemento]
    somma=sum(finestra)
    media=somma/len(finestra)
    print (finestra, somma, "media: ", media)
    x=input("**")
```

Avendo quindi invertito la lista, ho potuto lavorare su ogni elemento della lista creando un ciclo for che iniziasse dall'elemento 0 fino all'ultimo indice della lista che sarebbe la sua lunghezza.

Ho sfruttato il ciclo per poter creare una "finestra" di n numeri che potesse scorrere per ogni elemento della lista. Quindi ho usato la variabile "elemento" per poter aumentare il primo indice della lista "finestra" e l'n-esimo indice finale della finestra.

```
GNU nano 8.1      [esercizio252.py]
import os
os.system('clear')

lista=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
n=3
reversedLista=lista[::-1]

print ("la lista: ", lista, "e' lunga: ", len(lista))
print ("la lista al contrario e' ", reversedLista)
print ("n= ", n)

for elemento in range(0, len(reversedLista)):
    finestra=reversedLista[elemento:n+elemento]
    somma=sum(finestra)
    media=somma/len(finestra)
    print (finestra, somma, "media: ", media)
    x=input("**")
```

Infine, ho fatto tutte le operazioni necessarie, quindi la somma e la media sfruttando la somma, mandando tutto in output e mettendo un input vuoti (x=input("")) per monitorare tutto:

```
GNU nano 8.1 Esercizio2S2.py
import os
os.system('clear')

lista=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
n=3
reversedLista=lista[::-1]

print ("la lista: ", lista, "e' lunga: ", len(lista))
print ("la lista al contrario e' ", reversedLista)
print ("n= ", n)

for elemento in range(0, len(reversedLista)):
    finestra=reversedLista[elemento:n+elemento]
    somma=sum(finestra)
    media=somma/len(finestra)
    print (finestra, somma, "media: ", media)
    x=input("")
```

Output:

```
la lista: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] e' lunga: 10
la lista al contrario e' [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
n= 3
[10, 9, 8] 27 media: 9.0

[9, 8, 7] 24 media: 8.0

[8, 7, 6] 21 media: 7.0

[7, 6, 5] 18 media: 6.0

[6, 5, 4] 15 media: 5.0

[5, 4, 3] 12 media: 4.0

[4, 3, 2] 9 media: 3.0

[3, 2, 1] 6 media: 2.0

[2, 1] 3 media: 1.5

[1] 1 media: 1.0
```

## Numero 2

In questo esercizio ho inizialmente chiesto all'utente di inserire una frase in input che ho successivamente trasformato in lowercase (minuscolo), rimosso tutta la punteggiatura (quei caratteri dentro la parentesi sono standard per poter manipolare i testi su python) e infine ho splittato tutte le parole:

```
GNU nano 8.1 wordCounter.py
import re

testo=input("inserisci il testo: ")

testo=testo.lower() #trasforma il testo in minuscolo
testo=re.sub(r'^\w\s', '', testo) #rimuovi la punteggiatura
parole=testo.split()

conteggio={} #inizializzo il dizionario

for parola in parole:
    if parola in conteggio:
        conteggio[parola] += 1
    else:
        conteggio[parola] = 1

print (conteggio)
```

Successivamente ho inizializzato il dizionario vuoto che verrà riempito successivamente. Usando un ciclo for, per ogni parola della frase, si controlla se la parola è già dentro il dizionario. Se è già dentro il dizionario, si incrementa il suo contatore, mentre se non è presente, il suo contatore diventa 1.

```
GNU nano 8.1 wordCounter.py
import re

testo=input("inserisci il testo: ")

testo=testo.lower() #trasforma il testo in minuscolo
testo=re.sub(r'^\w\s', '', testo) #rimuovi la punteggiatura
parole=testo.split()

conteggio={} #inizializzo il dizionario

for parola in parole:
    if parola in conteggio:
        conteggio[parola] += 1
    else:
        conteggio[parola] = 1

print (conteggio)
```

Output:

```
(kali㉿kali)-[~/Documents/Epicode]  
$ python wordCounter.py  
inserisci il testo: Ciao, come stai? stai bene? come?  
{'ciao': 1, 'come': 2, 'stai': 2, 'bene': 1}
```