

Nel laboratorio di oggi, abbiamo svolto un esercizio che consisteva, con input e output.

Di seguito la consegna:

Richiesta di Input: Il programma deve chiedere all'utente di inserire:

- Il nome della città di origine.
- Il nome del proprio animale domestico.

Generazione del Nome della Band: Una volta ricevuti gli input, il programma deve combinare il nome della città e il nome dell'animale in un'unica stringa che rappresenta il nome della band.

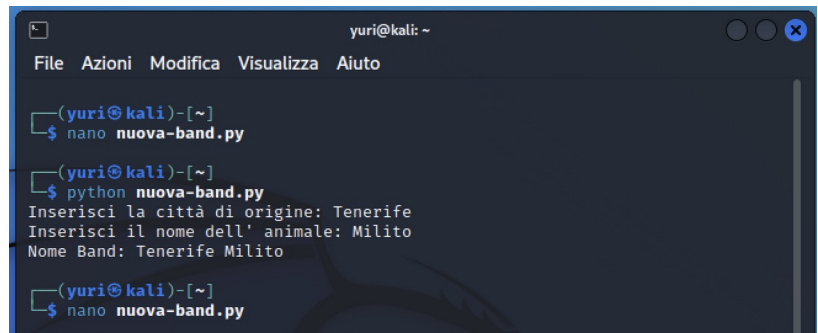
Output: Il programma deve stampare a video il nome generato per la band.

**Creazione delle variabili:** Ho creato due variabili, città e nome, e assegnato loro un valore tramite l'input dell'utente. La variabile città rappresenta la città di origine e nome rappresenta il nome dell'animale.



```
yuri@kali: ~  
File Azioni Modifica Visualizza Aiuto  
GNU nano 6.2 nuova-band.py  
Città = input("Inserisci la città di origine: ")  
Nome = input("Inserisci il nome dell' animale: ")  
print(f"Nome Band: {Città} {Nome}")
```

**Output delle variabili:** Ho utilizzato il comando print per stampare i valori delle variabili e visualizzarli nell'output, in modo che l'utente possa vedere cosa è stato inserito.



```
(yuri@kali)-[~]  
$ nano nuova-band.py  
  
(yuri@kali)-[~]  
$ python nuova-band.py  
Inserisci la città di origine: Tenerife  
Inserisci il nome dell' animale: Milito  
Nome Band: Tenerife Milito  
  
(yuri@kali)-[~]  
$ nano nuova-band.py
```

**Concatenazione delle variabili:** Ho creato una nuova variabile nome\_band, che concatena il valore di città e nome, creando così una stringa formata dai due input separati da un punto.

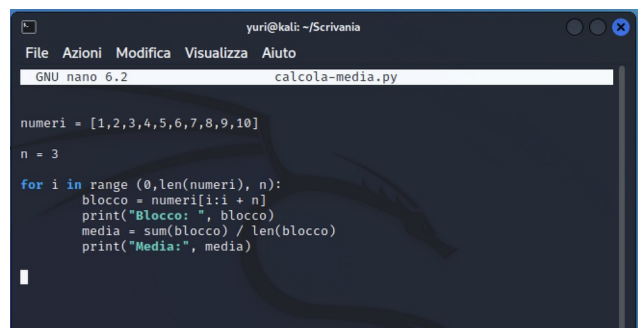
**Prova del programma:** Ho testato il programma inserendo una città e un nome come input per verificare che l'output fosse corretto.

Nel secondo esercizio, la consegna era la seguente:

Scrivi una funzione che calcoli la media mobile di una lista di numeri. La media mobile di un elemento è definita come la media degli ultimi n elementi della lista, inclusi l'elemento corrente.

ho provato un algoritmo che suddivide la lista in blocchi di lunghezza n, calcolando la media per ciascun blocco e restituendo l'elenco delle medie.

Questa funzione è stata testata utilizzando una lista di numeri dall'1 al 10, con n impostato a 3, per verificare il corretto funzionamento e l'output delle medie mobili.



```
yuri@kali: ~/Scrivania  
File Azioni Modifica Visualizza Aiuto  
GNU nano 6.2 calcola-media.py  
  
numeri = [1,2,3,4,5,6,7,8,9,10]  
n = 3  
for i in range(0, len(numeri), n):  
    blocco = numeri[i:i + n]  
    print("Blocco:", blocco)  
    media = sum(blocco) / len(blocco)  
    print("Media:", media)
```

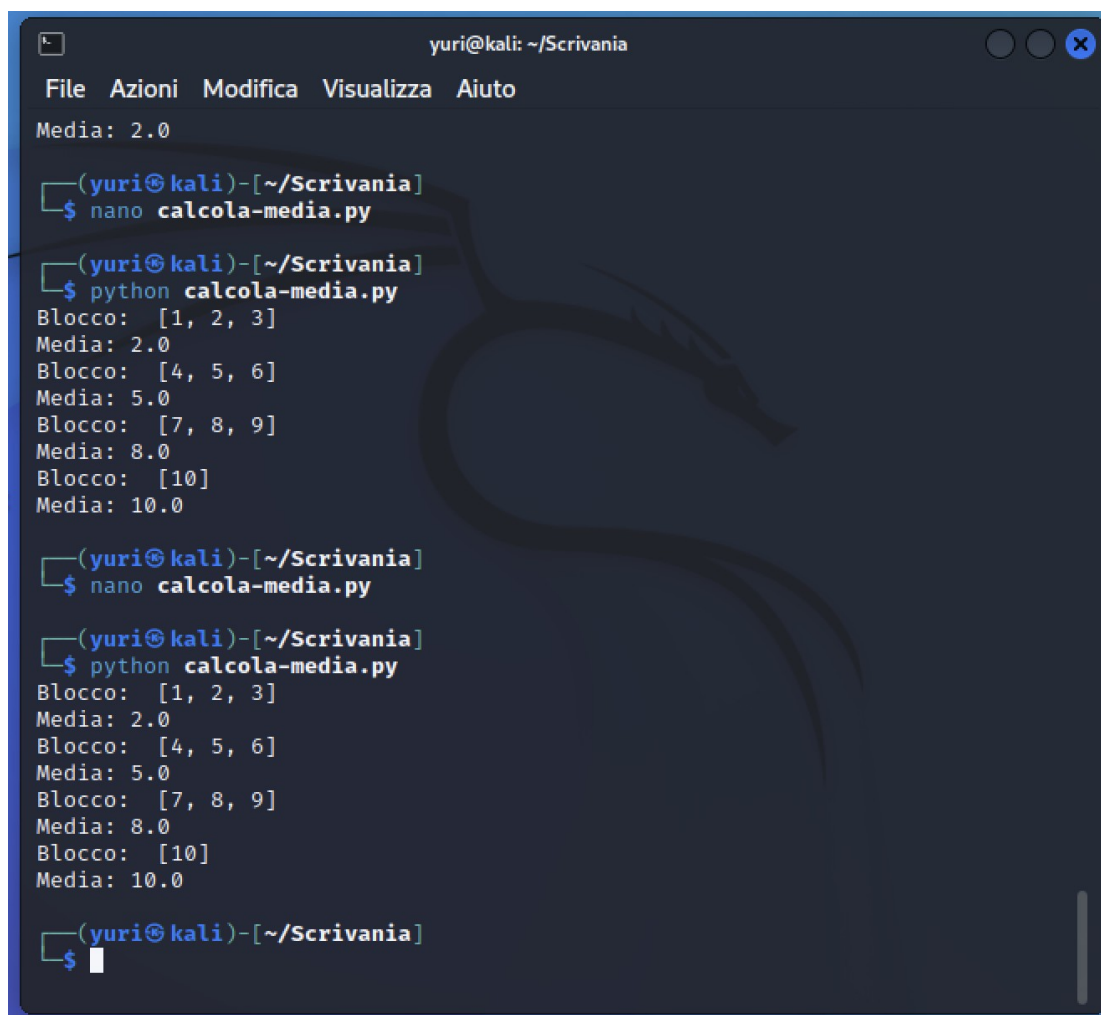
**Creazione dell'array di numeri:** Ho creato un array chiamato numeri contenente i numeri dall'1 al 10.

**Definizione della variabile n:** Ho creato una variabile n e le ho assegnato il valore 3. Questa variabile rappresenta la dimensione dei blocchi in cui voglio suddividere l'array.

**Suddivisione in blocchi:** Ho utilizzato un ciclo for per iterare attraverso l'array numeri e creare blocchi di 3 elementi ciascuno. Questo è stato fatto usando la notazione numeri[i:i+n], dove i è l'indice iniziale e i+n rappresenta l'indice finale per ciascun blocco.

**Stampa dei blocchi:** Ho stampato ogni blocco per mostrare visivamente i numeri raggruppati.

**Calcolo della media:** Per ciascun blocco, ho calcolato la media dei numeri contenuti al suo interno utilizzando la formula  $\text{sum}(\text{blocco}) / \text{len}(\text{blocco})$ , e ho stampato il risultato per mostrare la media di ciascun blocco.



```
yuri@kali: ~/Scrivania
File Azioni Modifica Visualizza Aiuto
Media: 2.0

(yuri@kali)-[~/Scrivania]
$ nano calcola-media.py

(yuri@kali)-[~/Scrivania]
$ python calcola-media.py
Blocco: [1, 2, 3]
Media: 2.0
Blocco: [4, 5, 6]
Media: 5.0
Blocco: [7, 8, 9]
Media: 8.0
Blocco: [10]
Media: 10.0

(yuri@kali)-[~/Scrivania]
$ nano calcola-media.py

(yuri@kali)-[~/Scrivania]
$ python calcola-media.py
Blocco: [1, 2, 3]
Media: 2.0
Blocco: [4, 5, 6]
Media: 5.0
Blocco: [7, 8, 9]
Media: 8.0
Blocco: [10]
Media: 10.0

(yuri@kali)-[~/Scrivania]
$
```