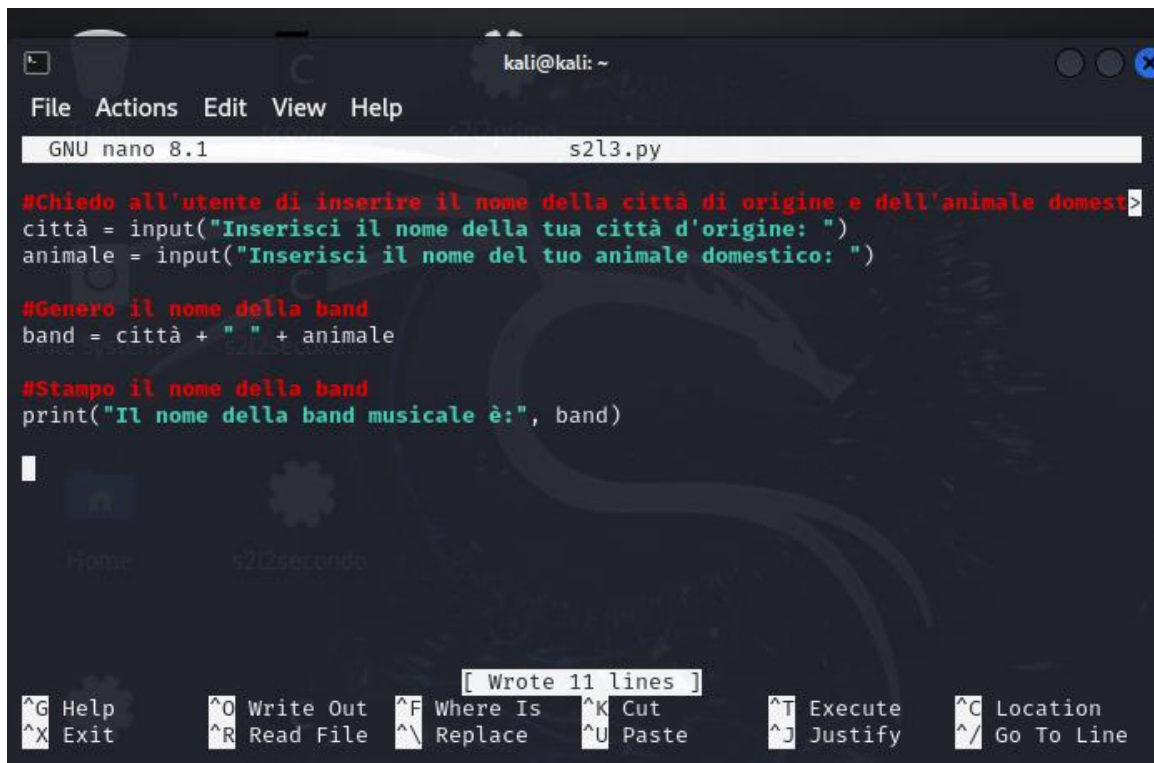


## Pratica S2/L3

- 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



```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 8.1 s2l3.py  
  
#Chiedo all'utente di inserire il nome della città di origine e dell'animale domestico  
città = input("Inserisci il nome della tua città d'origine: ")  
animale = input("Inserisci il nome del tuo animale domestico: ")  
  
#Genero il nome della band  
band = città + " " + animale  
  
#Stampo il nome della band  
print("Il nome della band musicale è:", band)  
  
[ Wrote 11 lines ]  
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```



```
kali@kali: ~  
File Actions Edit View Help  
- (kali@kali)-[~]  
$ touch s2l3.py  
- (kali@kali)-[~]  
$ nano test1.py  
- (kali@kali)-[~]  
$ nano s2l3.py  
- (kali@kali)-[~]  
$ python s2l3.py  
Inserisci il nome della tua città d'origine: New York  
Inserisci il nome del tuo animale domestico: Pimpolo  
Il nome della band musicale è: New York Pimpolo  
- (kali@kali)-[~]  
$ nano s2l3.py  
- (kali@kali)-[~]  
$ python s2l3.py  
Inserisci il nome della tua città d'origine: Pechino  
Inserisci il nome del tuo animale domestico: Ringhio  
Il nome della band musicale è: Pechino Ringhio
```

## Pratica bonus

### Esercizio 1

- Descrizione: Scrivi una funzione che calcoli la media mobile di una lista di numeri.

```
kali@kali: ~
File Actions Edit View Help
GNU nano 8.1 s2l3bonus1.py
def calcolo_media_mobile(numeri, n):
    medie_mobili = []
    for i in range(len(numeri) - n + 1):
        n_numeri = numeri[i:i + n]
        somma = sum(n_numeri)
        media = somma/n
        medie_mobili.append(media)
    return medie_mobili

numeri = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
n = 3
medie = calcolo_media_mobile(numeri, n)

print("La media mobile con finestra di dimensione", n, "è:", medie)
```

^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^\_ Go To Line

```
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ nano s2l3bonus1.py
(kali@kali)-[~]
$ pyt
Command 'pyt' not found, did you mean:
  command 'pt' from deb tcclib
  command 'py' from deb pythonpy
  command 'yyt' from deb yiyantang
  command 'prt' from deb prt
  command 'pat' from deb dist
  command 'pyp' from deb pyp
  command 'pyct' from deb python3-pyct
Try: sudo apt install <deb name>
(kali@kali)-[~]
$ python s2l3bonus1.py
La media mobile con finestra di dimensione 3 è: [2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0]
(kali@kali)-[~]
$
```

## Esercizio 2

- Scrivi una funzione che analizzi una stringa di testo e restituisca un dizionario con il conteggio delle occorrenze di ciascuna parola. Ignora la punteggiatura e considera le parole in modo case-insensitive.

```
GNU nano 8.1 s2l3bonus2.py
import re

def analizza_testo(testo):
    #Conversione del testo in minuscolo, per considerare le parole in modo case-insensitive
    testo = testo.lower()

    #rimozione della punteggiatura, tramite re
    testo = re.sub(r"[^\w\s]", " ", testo)

    #suddivisione stringa di testo in parole
    parole = testo.split()

    #creazione dizionario per tracciare occorrenze delle parole
    conteggio_parole = {}

    #conteggio delle occorrenze
    for parola in parole:
        if parola in conteggio_parole:
            conteggio_parole[parola] += 1
        else:
            conteggio_parole[parola] = 1

    return (conteggio_parole)

testo = "Stringa di prova, per verificare che questa prova funzioni!"
risultato = analizza_testo(testo)

print(risultato)
```

```
(kali@kali)-[~]
$ nano s2l3bonus2.py

(kali@kali)-[~]
$ python s2l3bonus2.py
{'stringa': 1, 'di': 1, 'prova': 2, 'per': 1, 'verificare': 1, 'che': 1, 'questa': 1, 'funzioni': 1}

(kali@kali)-[~]
$
```