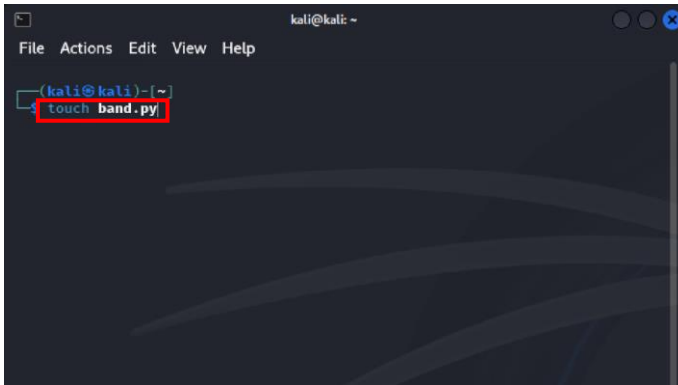


REPORT S2/L3

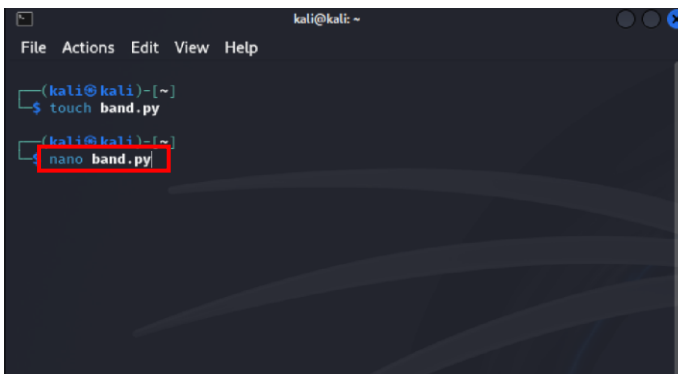
Scrivere un programma in **python** che genera il nome di una band composto dal nome di una città e dal nome di un animale domestico inseriti in **input** dall'utente.

Per prima cosa andremo a generare il nostro file band.py con il comando touch nome_file.py.

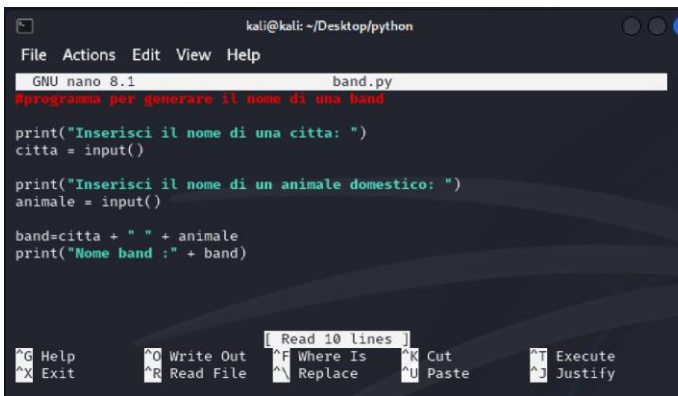


```
kali@kali: ~  
File Actions Edit View Help  
$ touch band.py
```

Successivamente lo andremo a scrivere con l'editor nano.

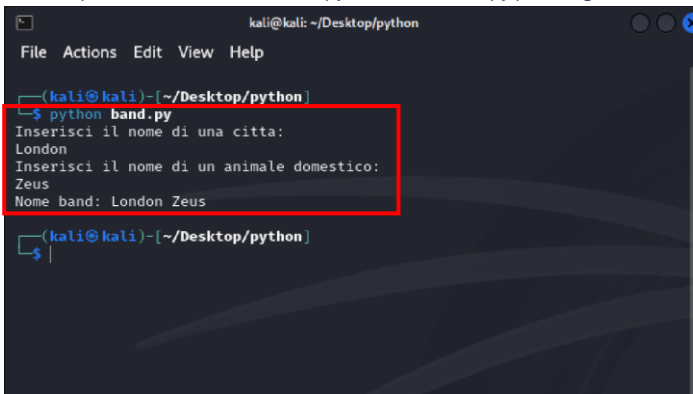


```
kali@kali: ~  
File Actions Edit View Help  
$ touch band.py  
$ nano band.py
```



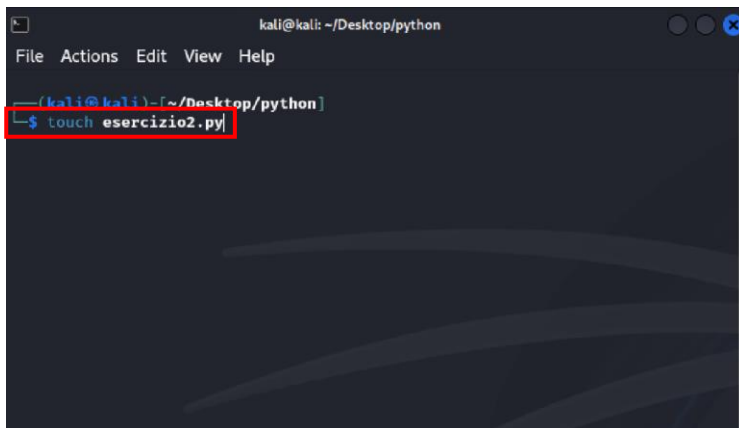
```
kali@kali: ~/Desktop/python  
File Actions Edit View Help  
GNU nano 8.1 band.py  
#programma per generare il nome di una band  
  
print("Inserisci il nome di una città: ")  
citta = input()  
  
print("Inserisci il nome di un animale domestico: ")  
animale = input()  
  
band=citta + " " + animale  
print("Nome band : " + band)
```

Infine si procederà con il comando python nome_file.py per eseguire il codice.



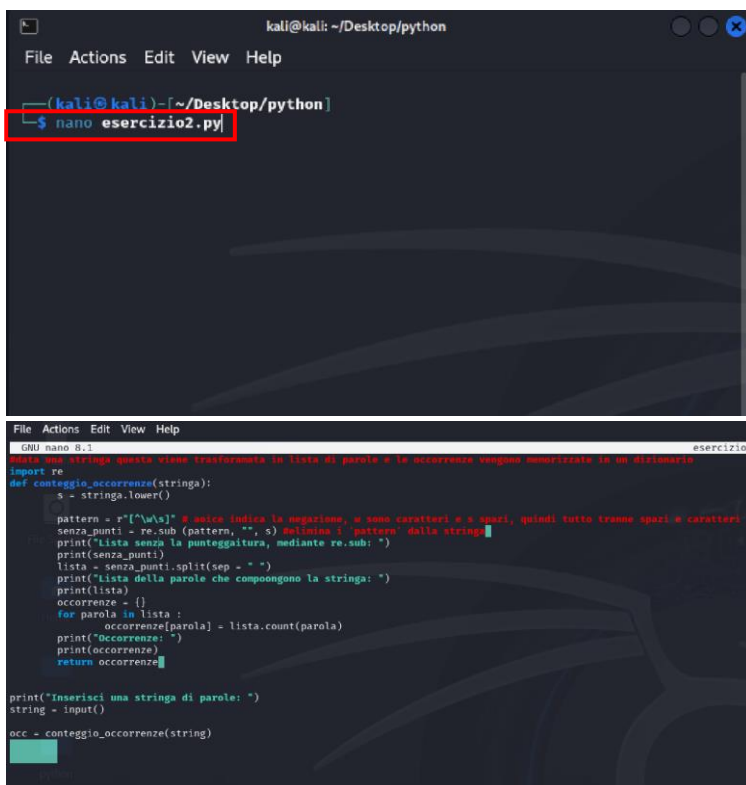
```
kali@kali: ~/Desktop/python  
File Actions Edit View Help  
$ python band.py  
Inserisci il nome di una città:  
London  
Inserisci il nome di un animale domestico:  
Zeus  
Nome band: London Zeus  
$
```

ESERCIZIO 2: Funzione che analizza una stringa di testo e restituisce un dizionario con il conteggio delle occorrenze di ciascuna parola. Ignora la punteggiatura e considera le parole in modo case-insensitive. Per prima cosa andremo a generare il nostro file **esercizio2.py** con il comando **touch nome_file.py**.



```
kali@kali: ~/Desktop/python
File Actions Edit View Help
--(kali@kali)~[~/Desktop/python]
$ touch esercizio2.py
```

Andremo poi ad aprire l' editor **nano**, mediante il comando **nano nome_file.py**

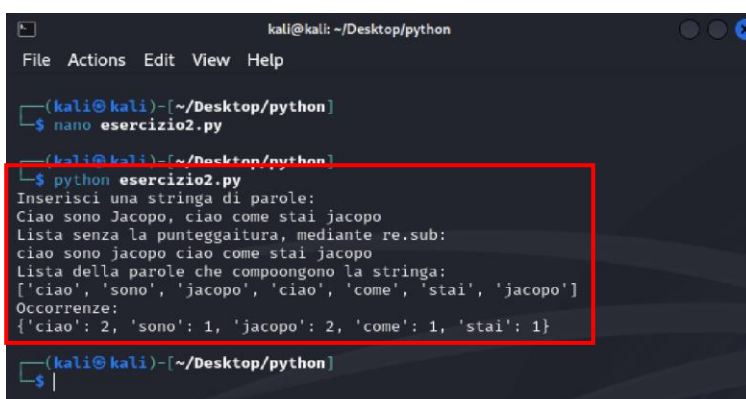


```
kali@kali: ~/Desktop/python
File Actions Edit View Help
--(kali@kali)~[~/Desktop/python]
$ nano esercizio2.py
```

```
GNU nano 8.1 esercizio2.py
#Questa stringa viene trasformata in lista di parole e le occorrenze vengono memorizzate in un dizionario
import re
def conteggio_occorrenze(stringa):
    s = stringa.lower()
    pattern = r"[^\w\s]" # la \w indica la negazione, e \s sono caratteri e \s spazi, quindi tutto tranne spazi e caratteri
    senza_punti = re.sub(pattern, "", s) #elimino i 'pattern' dalla stringa
    print("Lista senza la punteggiatura, mediante re.sub: ")
    print(senza_punti)
    lista = senza_punti.split(" ")
    print("Lista delle parole che compongono la stringa: ")
    print(lista)
    occorrenze = {}
    for parola in lista:
        occorrenze[parola] = lista.count(parola)
    print("Occorrenze: ")
    print(occorrenze)
    return occorrenze

print("Inserisci una stringa di parole: ")
string = input()
occ = conteggio_occorrenze(string)
```

Infine lo andremo ad eseguire mediante il comando **python nome_file.py**



```
kali@kali: ~/Desktop/python
File Actions Edit View Help
--(kali@kali)~[~/Desktop/python]
$ nano esercizio2.py
--(kali@kali)~[~/Desktop/python]
$ python esercizio2.py
Inserisci una stringa di parole:
Ciao sono Jacopo, ciao come stai jacopo
Lista senza la punteggiatura, mediante re.sub:
ciao sono jacopo ciao come stai jacopo
Lista delle parole che compongono la stringa:
['ciao', 'sono', 'jacopo', 'ciao', 'come', 'stai', 'jacopo']
Occorrenze:
{'ciao': 2, 'sono': 1, 'jacopo': 2, 'come': 1, 'stai': 1}
--(kali@kali)~[~/Desktop/python]
$
```