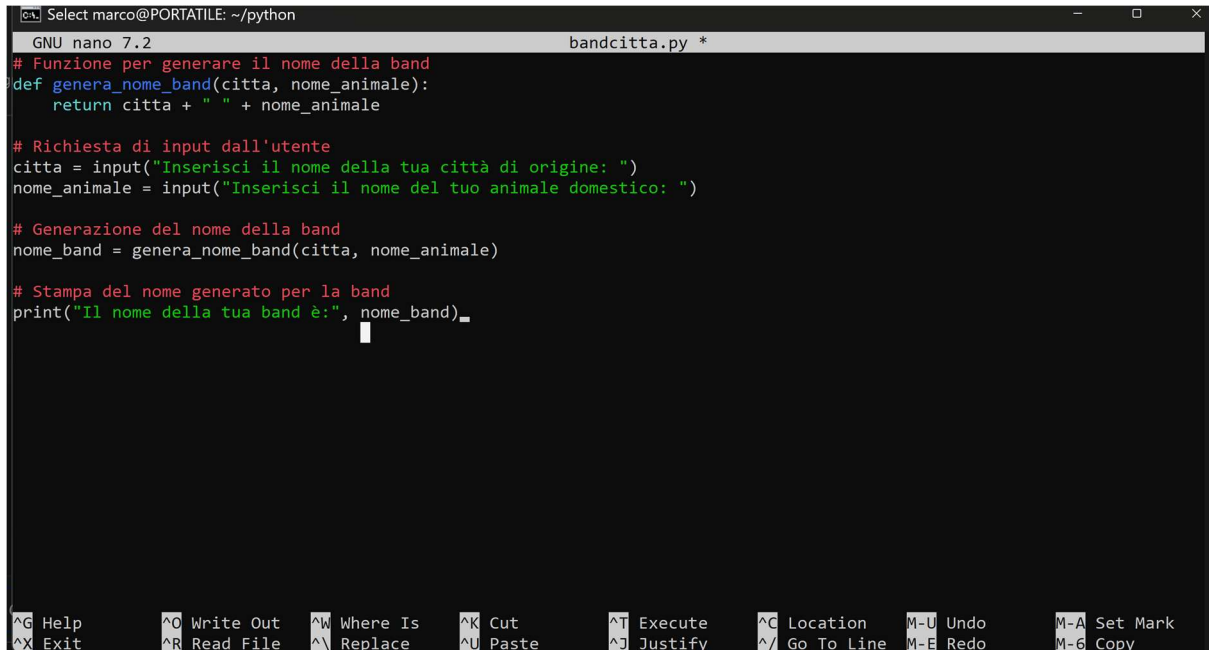


Per facilitare gli screenshot e la creazione del documento del report, userò WSL con ubuntu 24.04

Analogamente all'esercizio in C, creo la cartella e il file in cui inserire il codice.

A seguire il codice:



```
GNU nano 7.2 bandcitta.py *
# Funzione per generare il nome della band
def genera_nome_band(citta, nome_animale):
    return citta + " " + nome_animale

# Richiesta di input dall'utente
citta = input("Inserisci il nome della tua città di origine: ")
nome_animale = input("Inserisci il nome del tuo animale domestico: ")

# Generazione del nome della band
nome_band = genera_nome_band(citta, nome_animale)

# Stampa del nome generato per la band
print("Il nome della tua band è:", nome_band)
```

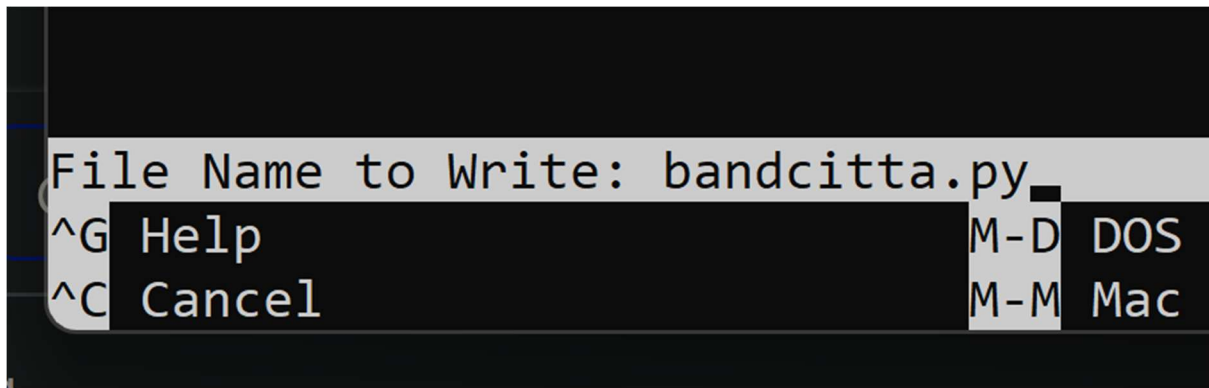
Ora analizziamo insieme il codice:

Come prima cosa viene definita la funzione **genera_nome_band** che ha in input due parametri: **citta** e **nome_animale**. La funzione combinerà i due parametri in un'unica stringa separata da uno spazio e la restituisce tramite **return**

Viene effettuata la richiesta di input da parte dell'utente. Viene richiesta la propria città di origine e il nome del proprio animale domestico. La funzione **input()** visualizza un messaggio all'utente e attende che l'utente inserisca un valore, che viene poi memorizzato nelle variabili **citta** e **nome_animale**.

Viene chiamata la funzione definita precedentemente che prende come input **citta** e **nome_animale**, il risultato viene salvato nella variabile **nome_band**

Viene infine stampato il nome della band tramite la funzione **print()**



Salvo il file e poi lo eseguo così:

```
marco@PORTATILE:~$ ls
python
marco@PORTATILE:~$ cd python/
marco@PORTATILE:~/python$ touch bandcitta.py
marco@PORTATILE:~/python$ nano bandcitta.py
marco@PORTATILE:~/python$ python3 bandcitta.py
Inserisci il nome della tua città di origine: Brescia
Inserisci il nome del tuo animale domestico: Jimmy
Il nome della tua band è: Brescia Jimmy
marco@PORTATILE:~/python$
```

Il codice funziona e l'output è corretto. Il codice può essere migliorato per gestire situazioni in cui l'input è vuoto come a seguire:

```
marco@PORTATILE:~/python$ python3 bandcitta.py
Inserisci il nome della tua città di origine:
Inserisci il nome del tuo animale domestico:
Il nome della tua band è:
marco@PORTATILE:~/python$
```

Per migliorare il codice aggiungiamo un controllo sull'input per evitare che l'input sia vuoto e per rimuovere tutti gli spazi all'inizio e alla fine delle stringhe che acquisiamo tramite input utente

```
marco@PORTATILE: ~/python
GNU nano 7.2 bandcittav2.py *
# Funzione per generare il nome della band
def genera_nome_band(citta, nome_animale):
    return citta + " " + nome_animale

# Funzione per richiedere input non vuoto
def richiedi_input(messaggio):
    return input(messaggio).strip()

# Richiesta di input dall'utente con controllo
citta = richiedi_input("Inserisci il nome della tua città di origine: ")
while not citta:
    print("L'input non può essere vuoto. Per favore, riprova.")
    citta = richiedi_input("Inserisci il nome della tua città di origine: ")

nome_animale = richiedi_input("Inserisci il nome del tuo animale domestico: ")
while not nome_animale:
    print("L'input non può essere vuoto. Per favore, riprova.")
    nome_animale = richiedi_input("Inserisci il nome del tuo animale domestico: ")

# Generazione del nome della band
nome_band = genera_nome_band(citta, nome_animale)

# Stampa del nome generato per la band
print("Il nome della tua band è:", nome_band)
```

La condizione `while not citta` viene utilizzata per verificare se la variabile `citta` è vuota. In Python, un valore vuoto (come una stringa vuota `""`) è considerato `False` in un contesto booleano. Quindi, `not citta` sarà `True` se `citta` è vuota.

Il ciclo `while not citta` continua a chiedere all'utente di inserire un valore finché non viene fornito un input non vuoto.

Questo garantisce che l'utente non possa procedere senza fornire un valore valido per `citta`. Lo stesso principio si applica per `nome_animale`.

La funzione **`.strip()`** in Python è un metodo delle stringhe che rimuove tutti gli spazi bianchi iniziali e finali dalla stringa. Questo è utile per pulire l'input dell'utente, eliminando eventuali spazi non necessari che potrebbero essere stati inseriti accidentalmente.