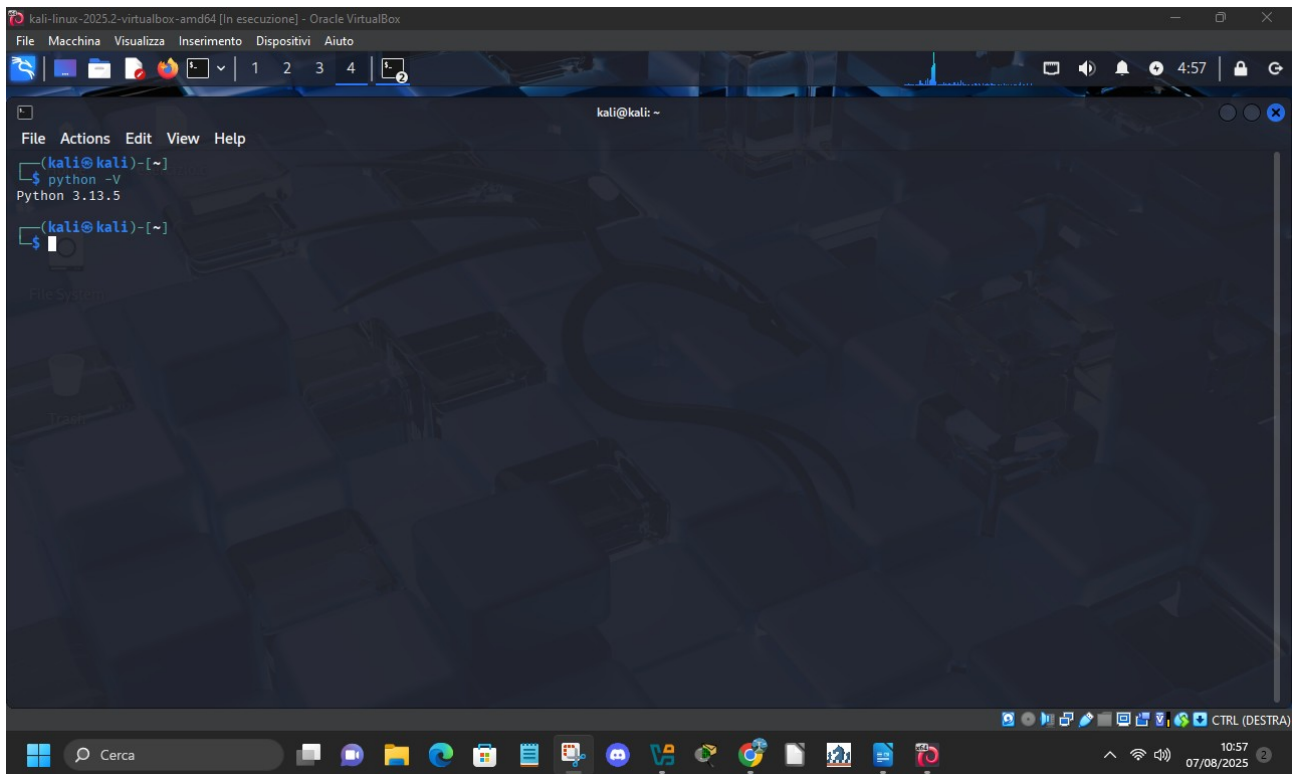


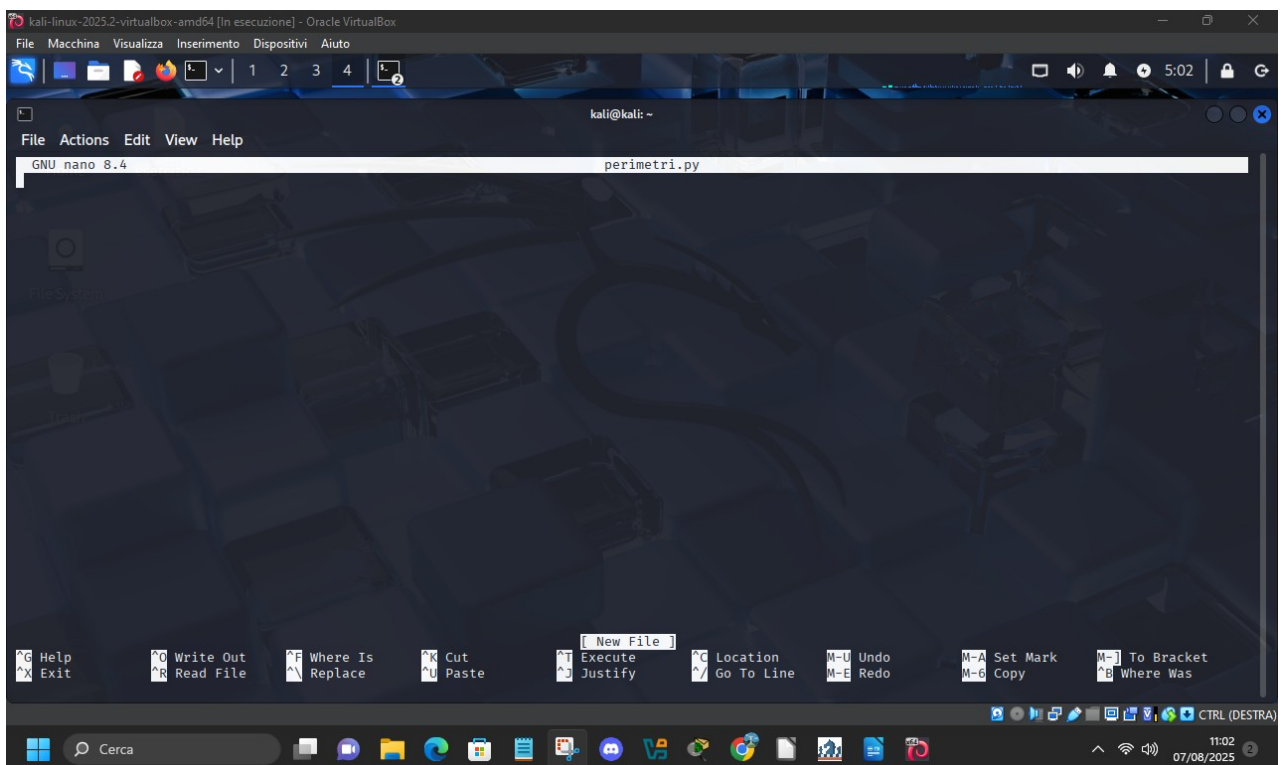
W6D4- FRANCESCO MONTALTO

Partiamo aprendo Kali Linux e verifichiamo che Python sia installato tramite il comando “python -V” dal terminale.



```
kali@kali: ~  
$ python -V  
Python 3.13.5
```

A questo punto creiamo un file Python tramite l’editor Nano, che chiameremo, nel nostro caso, “perimetri.py”, in modo da scrivere in codice python. Ci troveremo davanti a questa situazione:



```
GNU nano 8.4 perimetri.py
```

Importiamo il modulo math, necessaria per il calcolo della circonferenza del cerchio, che in python non è predefinita. La formula per calcolare la circonferenza del cerchio è:

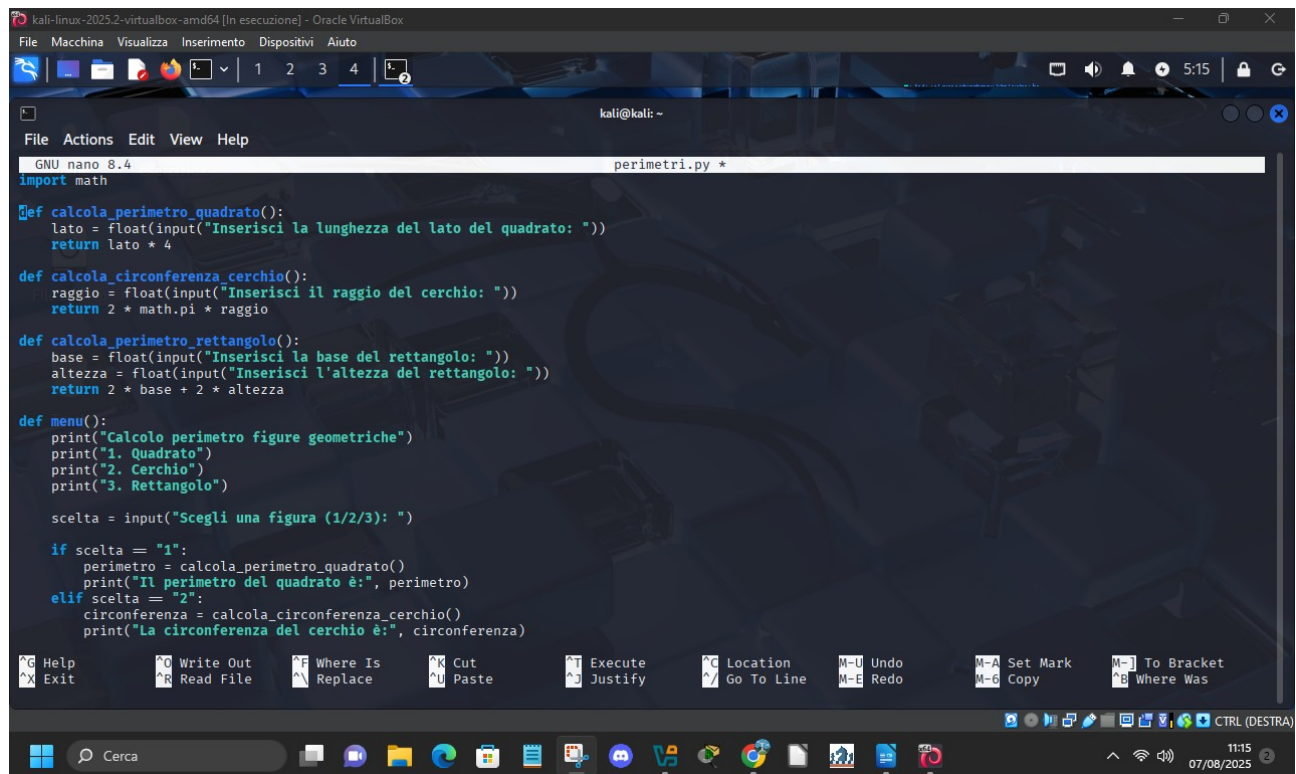
`"2 * math.pi * raggio"`

Per rendere il codice più ordinato abbiamo diviso il programma in funzioni; una funzione per ciascuna figura. Ogni funzione chiede un numero all'utente, lo calcola e ne ritorna il risultato.

Con `"input()"` chiediamo all'utente di scrivere un numero.

Con `"float()"` lo convertiamo in un numero vero, altrimenti resterebbe solo testo.

A fine programma, come si vede, abbiamo usato la funzione `"menu()"`, per stamparne i risultati.



```
GNU nano 8.4 perimetri.py *
import math

def calcola_perimetro_quadrato():
    lato = float(input("Inserisci la lunghezza del lato del quadrato: "))
    return lato * 4

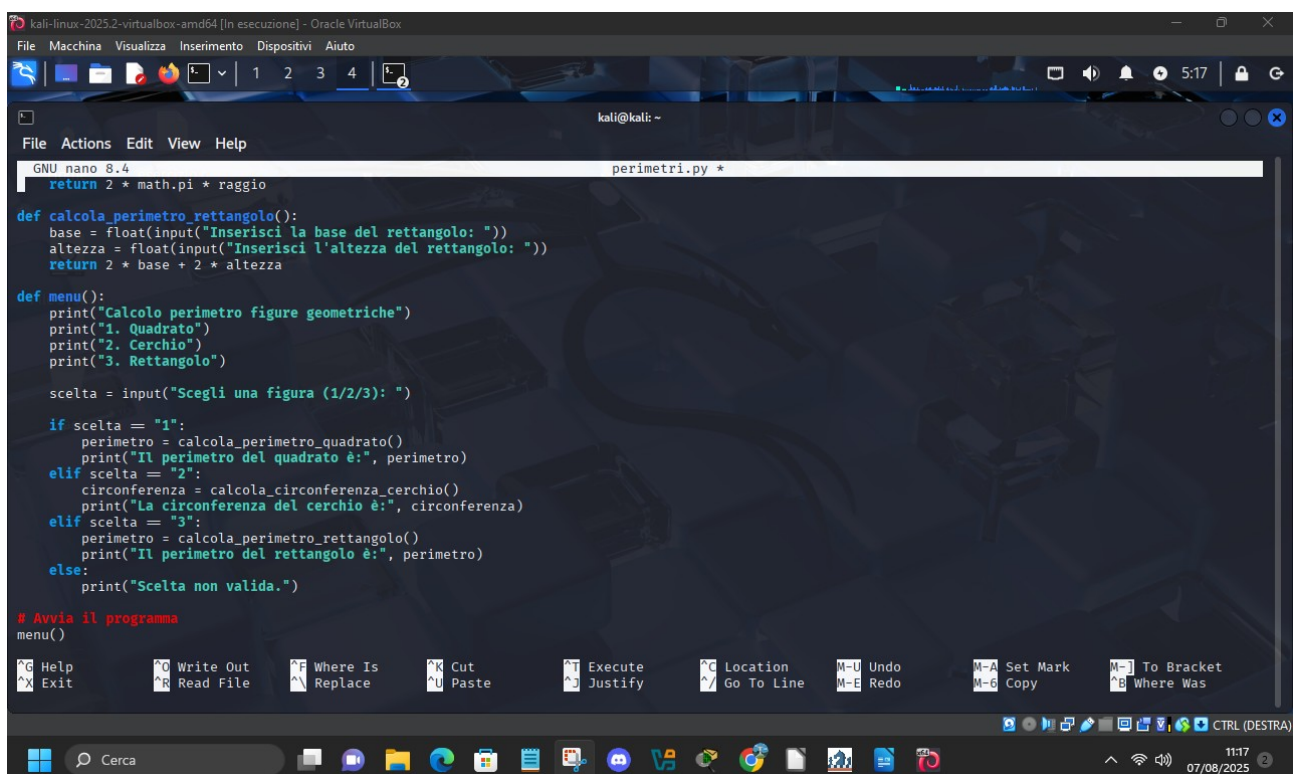
def calcola_circonferenza_cerchio():
    raggio = float(input("Inserisci il raggio del cerchio: "))
    return 2 * math.pi * raggio

def calcola_perimetro Rettangolo():
    base = float(input("Inserisci la base del rettangolo: "))
    altezza = float(input("Inserisci l'altezza del rettangolo: "))
    return 2 * base + 2 * altezza

def menu():
    print("Calcolo perimetro figure geometriche")
    print("1. Quadrato")
    print("2. Cerchio")
    print("3. Rettangolo")

    scelta = input("Scegli una figura (1/2/3): ")

    if scelta == "1":
        perimetro = calcola_perimetro_quadrato()
        print("Il perimetro del quadrato è:", perimetro)
    elif scelta == "2":
        circonferenza = calcola_circonferenza_cerchio()
        print("La circonferenza del cerchio è:", circonferenza)
```



```
GNU nano 8.4 perimetri.py *
    return 2 * math.pi * raggio

def calcola_perimetro Rettangolo():
    base = float(input("Inserisci la base del rettangolo: "))
    altezza = float(input("Inserisci l'altezza del rettangolo: "))
    return 2 * base + 2 * altezza

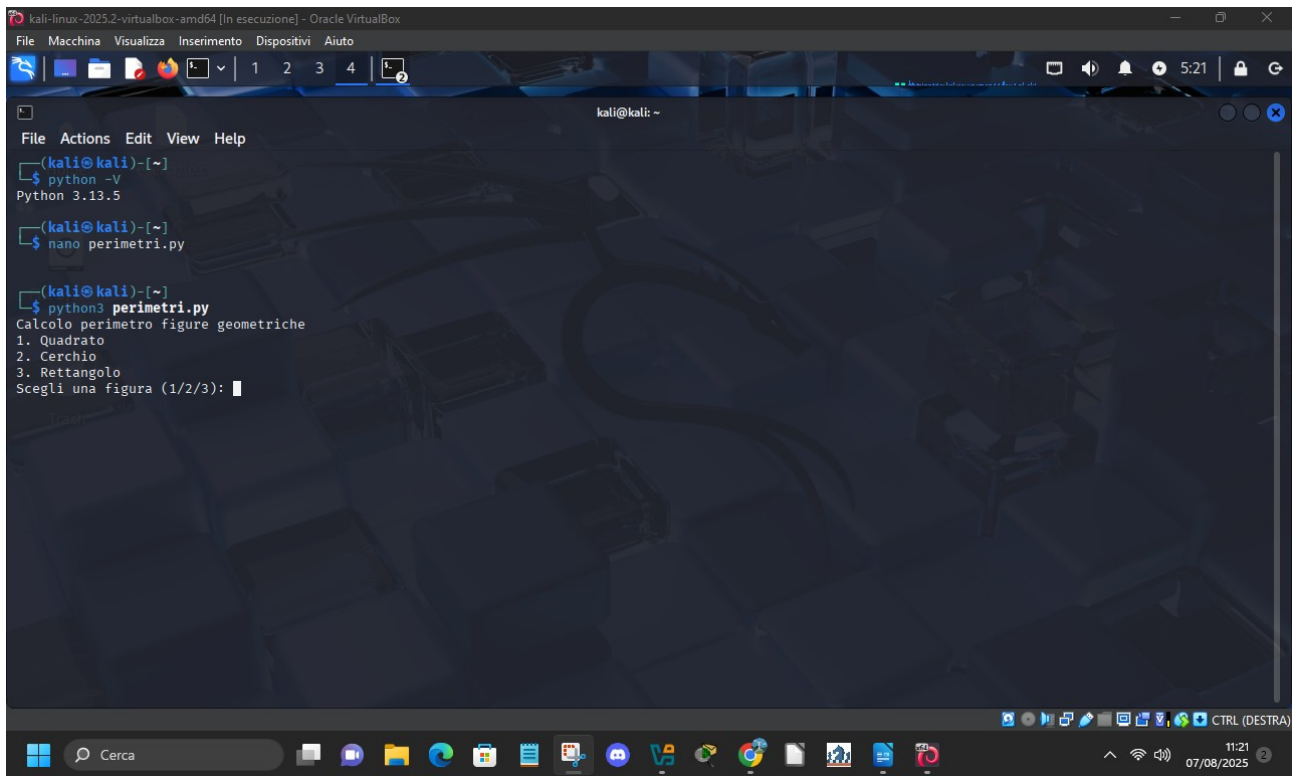
def menu():
    print("Calcolo perimetro figure geometriche")
    print("1. Quadrato")
    print("2. Cerchio")
    print("3. Rettangolo")

    scelta = input("Scegli una figura (1/2/3): ")

    if scelta == "1":
        perimetro = calcola_perimetro_quadrato()
        print("Il perimetro del quadrato è:", perimetro)
    elif scelta == "2":
        circonferenza = calcola_circonferenza_cerchio()
        print("La circonferenza del cerchio è:", circonferenza)
    elif scelta == "3":
        perimetro = calcola_perimetro Rettangolo()
        print("Il perimetro del rettangolo è:", perimetro)
    else:
        print("Sceita non valida.")

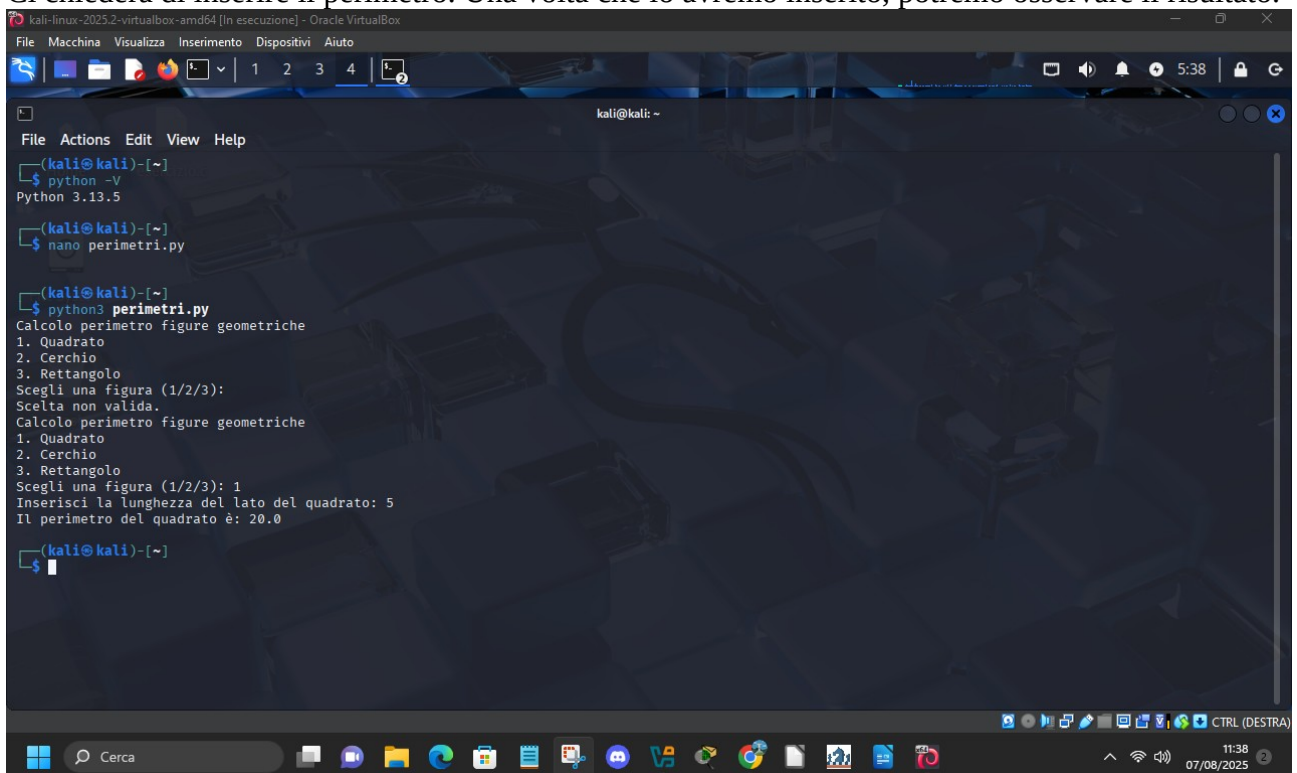
# Avvia il programma
menu()
```

Adesso occorre salvare ed uscire dal file (ctrl+o; invio, ctrl+x).
A questo punto avviamo semplicemente il file dal terminale, tramite il comando “python3 perimetri.py”.



```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ python -V  
Python 3.13.5  
(kali@kali)-[~]  
$ nano perimetri.py  
(kali@kali)-[~]  
$ python3 perimetri.py  
Calcolo perimetro figure geometriche  
1. Quadrato  
2. Cerchio  
3. Rettangolo  
Scegli una figura (1/2/3):
```

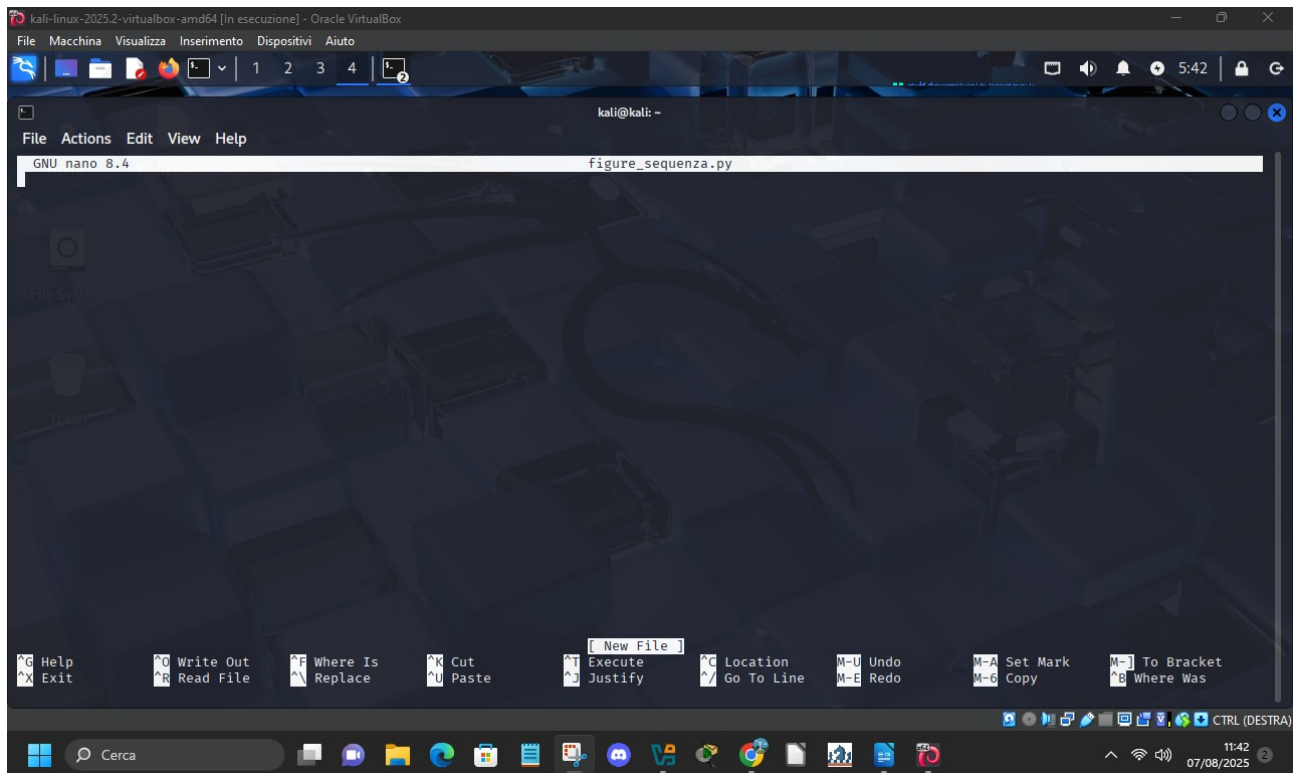
Ora scegliamo una figura e premiamo invio.
Ci chiederà di inserire il perimetro. Una volta che lo avremo inserito, potremo osservare il risultato.



```
(kali@kali)-[~]  
$ python3 perimetri.py  
Calcolo perimetro figure geometriche  
1. Quadrato  
2. Cerchio  
3. Rettangolo  
Scegli una figura (1/2/3): 1  
Sceita non valida.  
Calcolo perimetro figure geometriche  
1. Quadrato  
2. Cerchio  
3. Rettangolo  
Scegli una figura (1/2/3): 1  
Inserisci la lunghezza del lato del quadrato: 5  
Il perimetro del quadrato è: 20.0  
(kali@kali)-[~]  
$
```

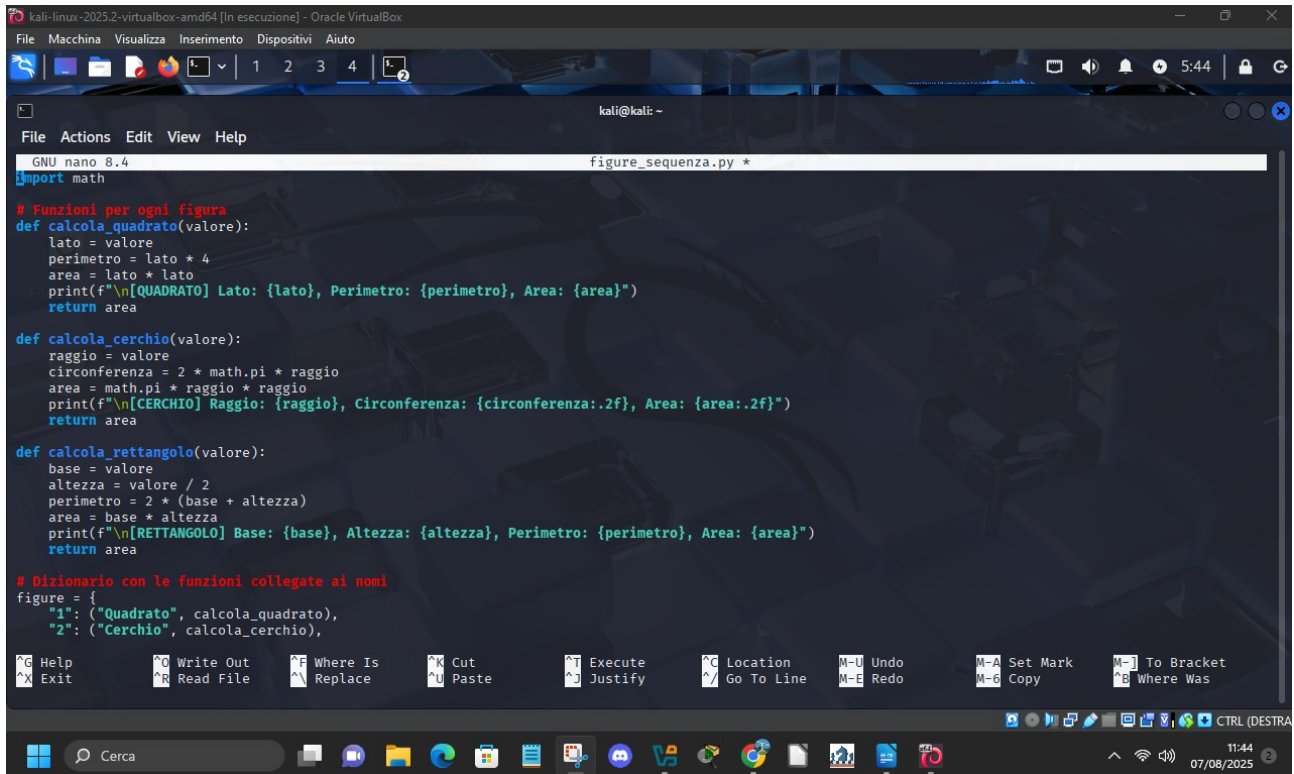
ESERCIZIO FACOLTATIVO:

Nel terminale scriviamo il comando “nano figure_sequenza.py” e si aprirà GNU.



A questo punto scriviamo i relativi comandi:

1. Come prima, descrizione del programma e funzionamento delle funzioni, in cui il quadrato riceve il lato, il cerchio il raggio, il rettangolo la base, e ne stampano i risultati.
2. Si inserisce un valore iniziale, si sceglie una figura da calcolare e si calcola perimetro e area usando il valore fornito.
3. L'area calcolata viene usata come nuovo valore per il calcolo dell'area successiva.
4. Il programma stamperà, al termine, un messaggio di completamento.



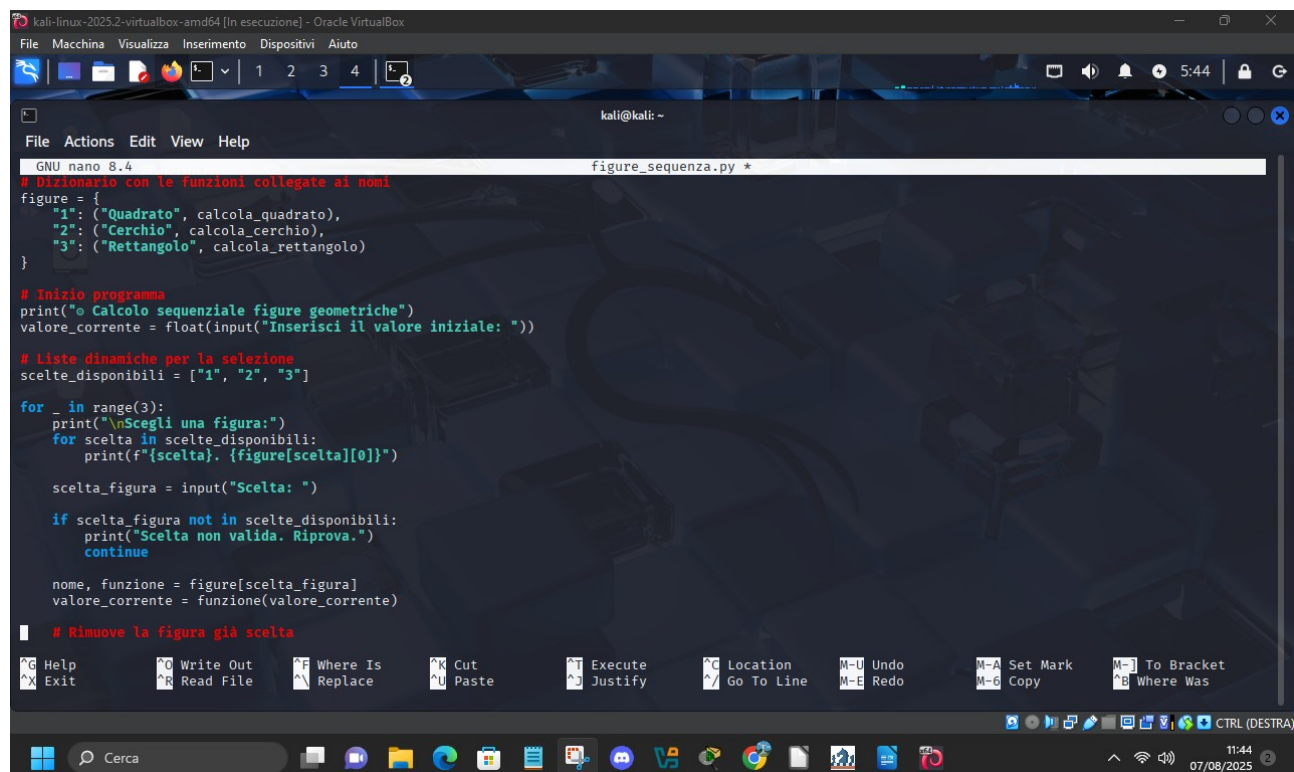
```
GNU nano 8.4 figure_sequenza.py *
import math

# Funzioni per ogni figura
def calcola_quadrato(valore):
    lato = valore
    perimetro = lato * 4
    area = lato * lato
    print(f"\n[QUADRATO] Lato: {lato}, Perimetro: {perimetro}, Area: {area}")
    return area

def calcola_cerchio(valore):
    raggio = valore
    circonferenza = 2 * math.pi * raggio
    area = math.pi * raggio * raggio
    print(f"\n[CERCHIO] Raggio: {raggio}, Circonferenza: {circonferenza:.2f}, Area: {area:.2f}")
    return area

def calcola Rettangolo(valore):
    base = valore
    altezza = valore / 2
    perimetro = 2 * (base + altezza)
    area = base * altezza
    print(f"\n[RETTANGOLO] Base: {base}, Altezza: {altezza}, Perimetro: {perimetro}, Area: {area}")
    return area

# Dizionario con le funzioni collegate ai nomi
figure = {
    "1": ("Quadrato", calcola_quadrato),
    "2": ("Cerchio", calcola_cerchio),
    "3": ("Rettangolo", calcola Rettangolo)
}
```



```
GNU nano 8.4 figure_sequenza.py *
# Dizionario con le funzioni collegate ai nomi
figure = {
    "1": ("Quadrato", calcola_quadrato),
    "2": ("Cerchio", calcola_cerchio),
    "3": ("Rettangolo", calcola Rettangolo)
}

# Inizio programma
print("\n Calcolo sequenziale figure geometriche")
valore_corrente = float(input("Inserisci il valore iniziale: "))

# Liste dinamiche per la selezione
scelte_disponibili = ["1", "2", "3"]

for _ in range(3):
    print("\nScegli una figura:")
    for scelta in scelte_disponibili:
        print(f"{scelta}. {figure[scelta][0]}")

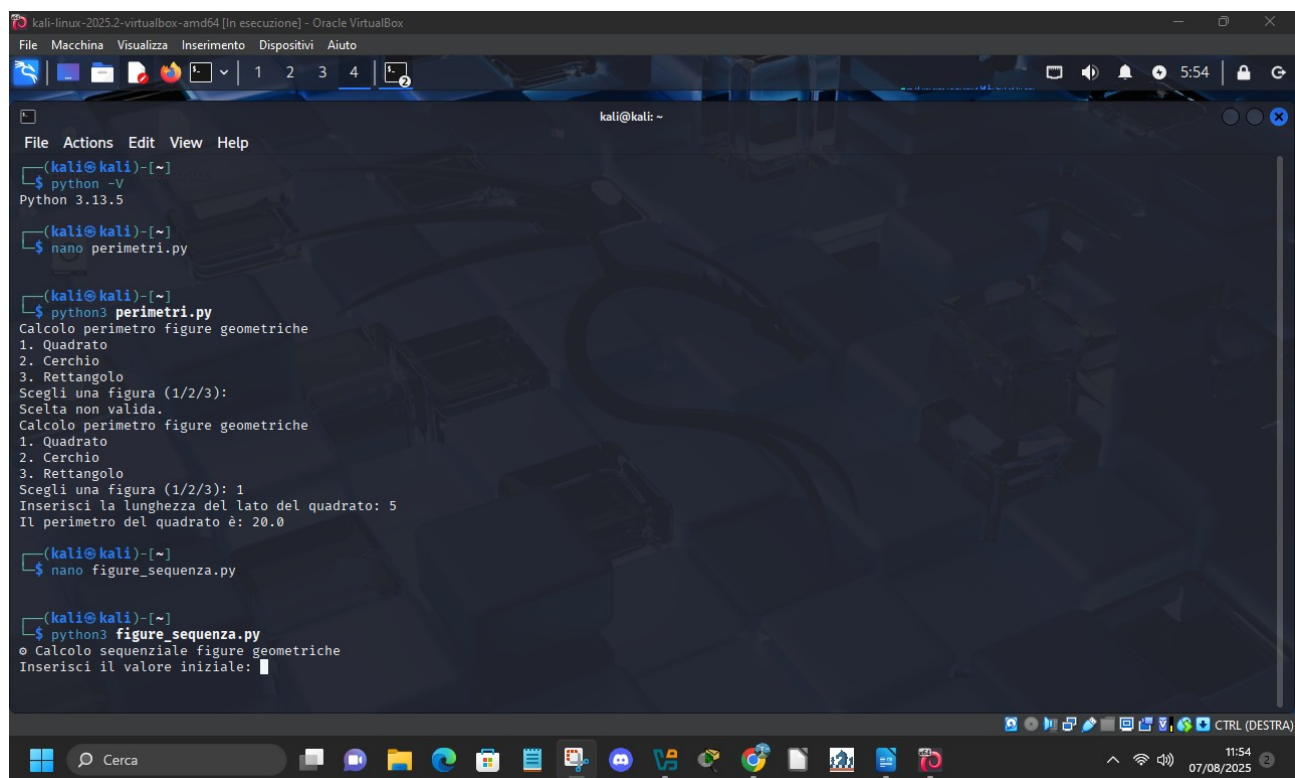
    scelta_figura = input("Scelta: ")

    if scelta_figura not in scelte_disponibili:
        print("Scelta non valida. Riprova.")
        continue

    nome, funzione = figure[scelta_figura]
    valore_corrente = funzione(valore_corrente)

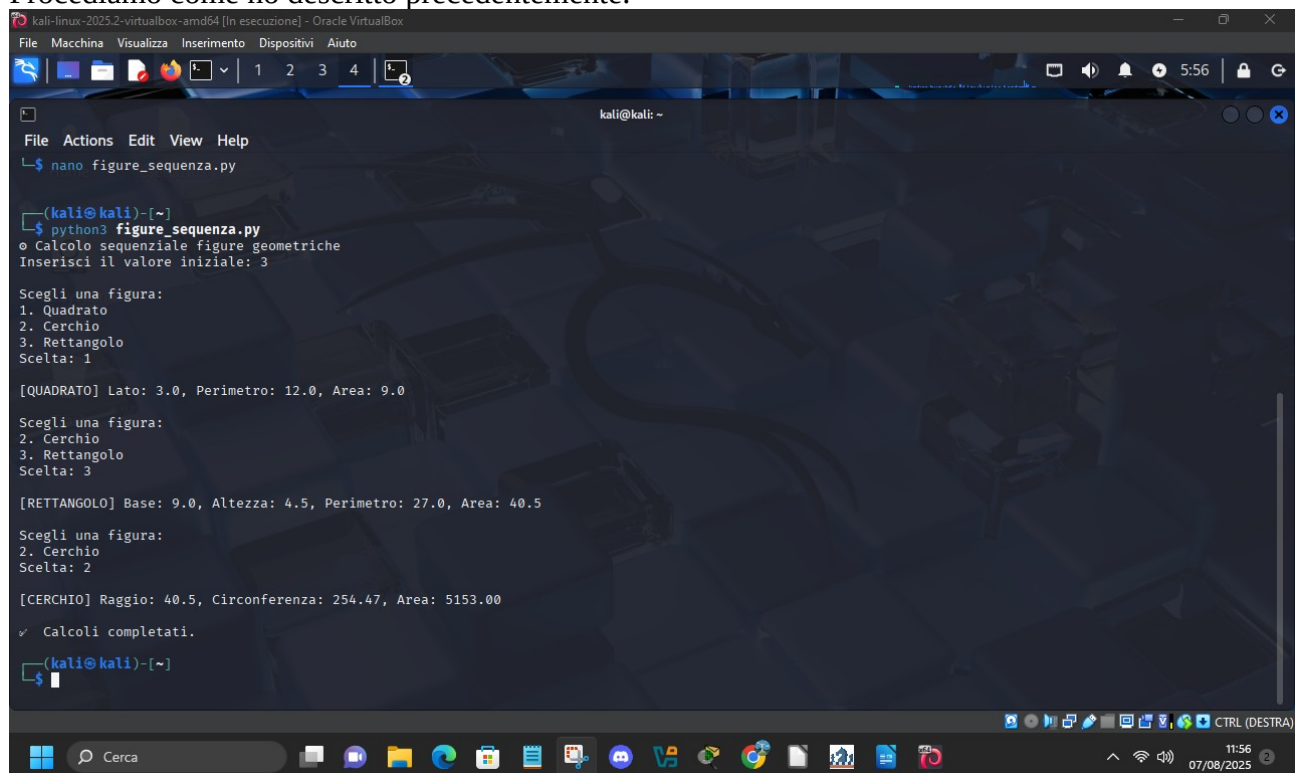
# Rimuove la figura già scelta
```

Salviamo il file e lo eseguiamo come “python3 figure_sequenza.py”.
Ci chiederà di inserire il valore iniziale per il calcolo sequenziale delle figure.



```
kali@kali: ~  
$ python -V  
Python 3.13.5  
$ nano perimetri.py  
$ python3 perimetri.py  
Calcolo perimetro figure geometriche  
1. Quadrato  
2. Cerchio  
3. Rettangolo  
Scegli una figura (1/2/3):  
Scelta non valida.  
Calcolo perimetro figure geometriche  
1. Quadrato  
2. Cerchio  
3. Rettangolo  
Scegli una figura (1/2/3): 1  
Inserisci la lunghezza del lato del quadrato: 5  
Il perimetro del quadrato è: 20.0  
$ nano figure_sequenza.py  
$ python3 figure_sequenza.py  
o Calcolo sequenziale figure geometriche  
Inserisci il valore iniziale: 
```

Procediamo come ho descritto precedentemente.



```
kali@kali: ~  
$ nano figure_sequenza.py  
$ python3 figure_sequenza.py  
o Calcolo sequenziale figure geometriche  
Inserisci il valore iniziale: 3  
  
Scegli una figura:  
1. Quadrato  
2. Cerchio  
3. Rettangolo  
Scelta: 1  
[QUADRATO] Lato: 3.0, Perimetro: 12.0, Area: 9.0  
  
Scegli una figura:  
2. Cerchio  
3. Rettangolo  
Scelta: 3  
[RETTANGOLO] Base: 9.0, Altezza: 4.5, Perimetro: 27.0, Area: 40.5  
  
Scegli una figura:  
2. Cerchio  
Scelta: 2  
[CERCHIO] Raggio: 40.5, Circonferenza: 254.47, Area: 5153.00  
  
✓ Calcoli completati.  
$ 
```

k