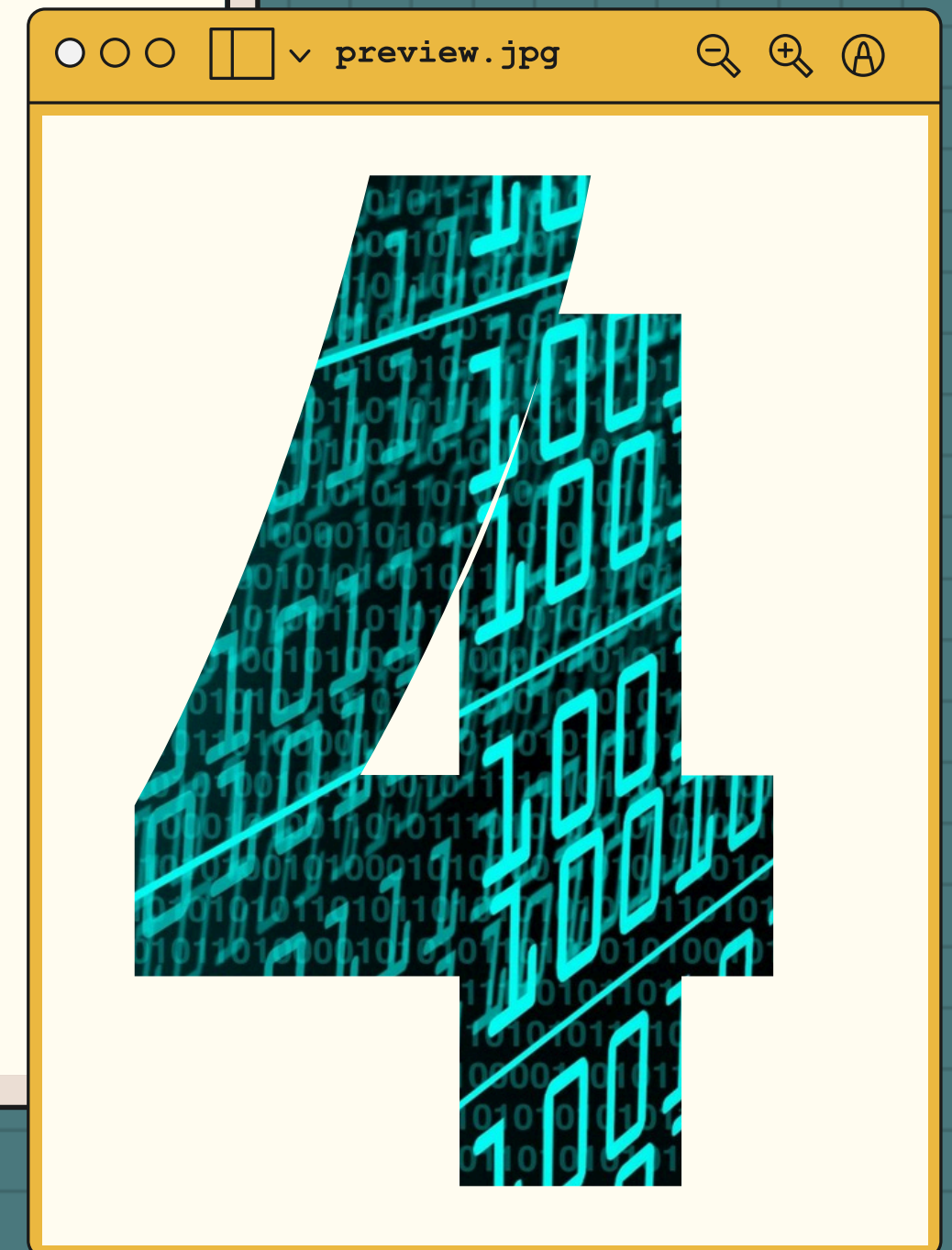


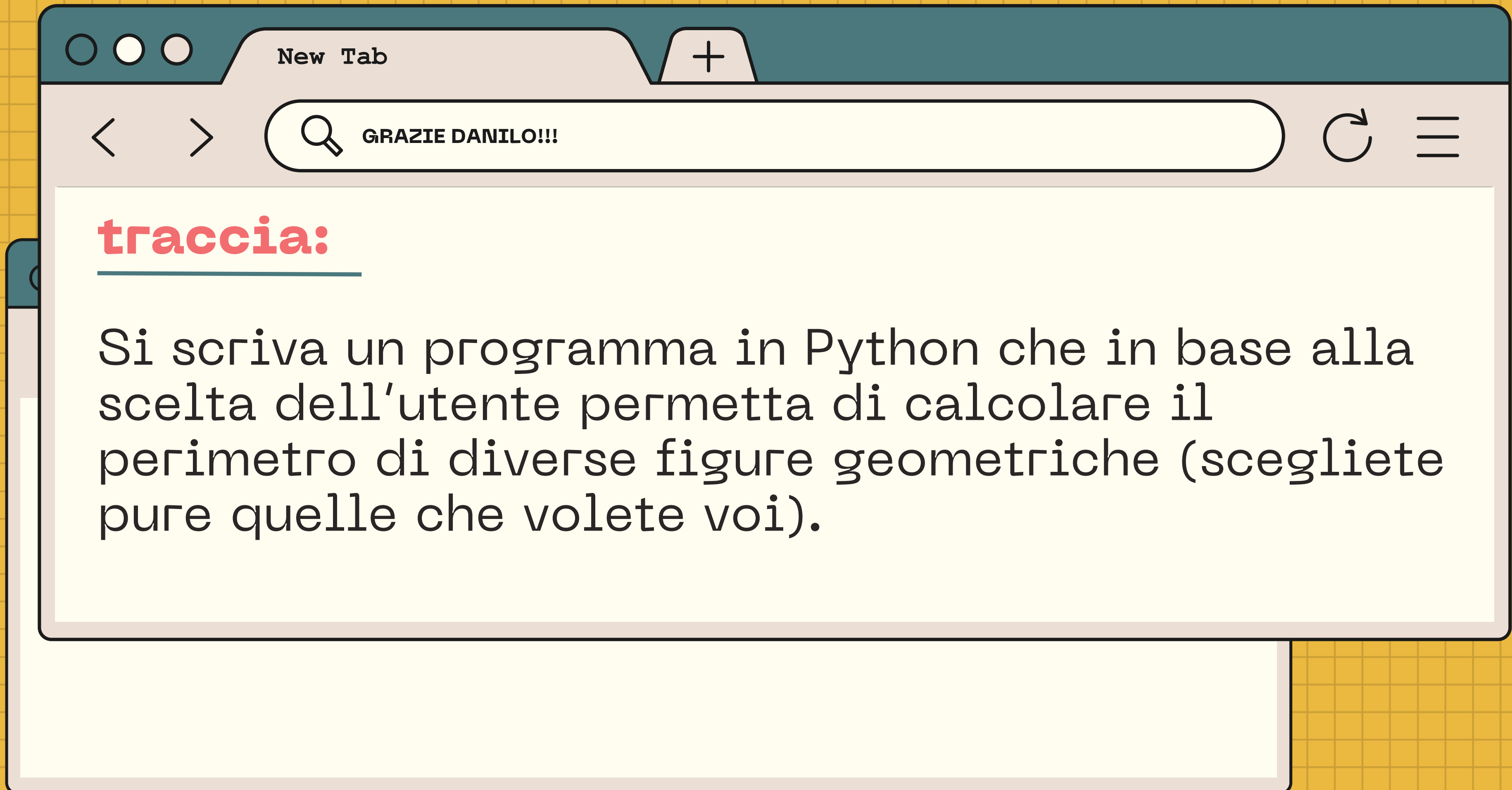


## TEAM 4

**Team leader:** Danilo Malagoli

**Member:** Matteo Tedesco, Alberto Guimp, Michele Covi, Joel Mouafo, Max Aldrovandi

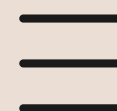




New Tab



GRAZIE DANILO!!!



## traccia:

Si scriva un programma in Python che in base alla scelta dell'utente permetta di calcolare il perimetro di diverse figure geometriche (scegliete pure quelle che volete voi).

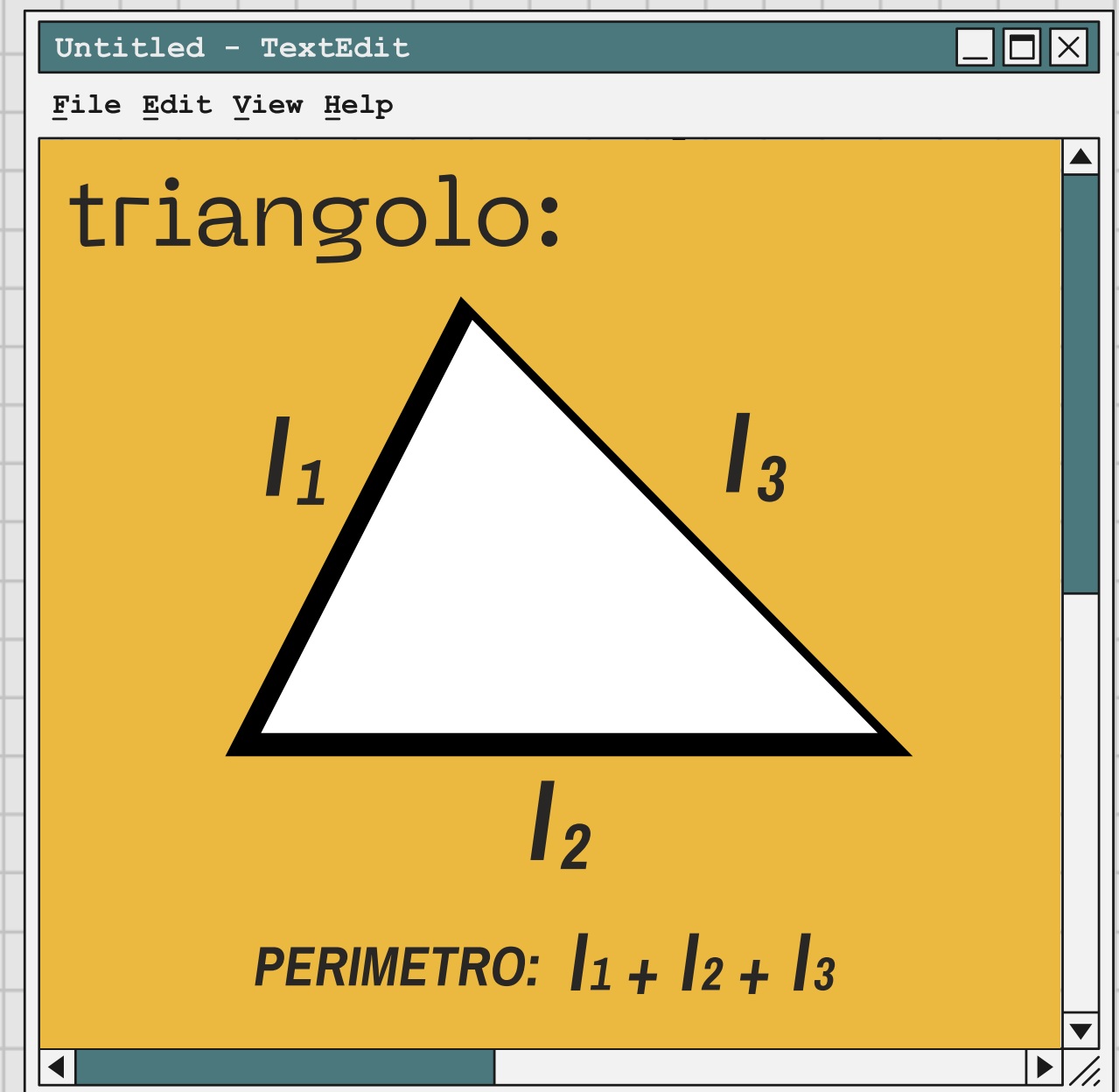
## codice Python:

```
def calcola_perimetro_triangolo_scaleno(lato_a, lato_b, lato_c):  
    return lato_a + lato_b + lato_c
```

```
elif scelta == '3':  
    lato_a = float(input("Inserisci il primo lato del triangolo scaleno: "))  
    lato_b = float(input("Inserisci il secondo lato del triangolo scaleno: "))  
    lato_c = float(input("Inserisci il terzo lato del triangolo scaleno: "))  
    print("Perimetro del triangolo scaleno:", calcola_perimetro_triangolo_scaleno(lato_a, lato_b, lato_c))
```

```
(kali@kali)-[~]  
└─$ python3 exercise.py  
Calcolatore di perimetro e circonferenza  
1. Parallelogramma  
2. Cerchio  
3. Triangolo scaleno  
Scegli una figura (1-3): 3  
Inserisci il primo lato del triangolo scaleno: 10  
Inserisci il secondo lato del triangolo scaleno: 23  
Inserisci il terzo lato del triangolo scaleno: 21  
Perimetro del triangolo scaleno: 54.0
```

## Figura scelta:



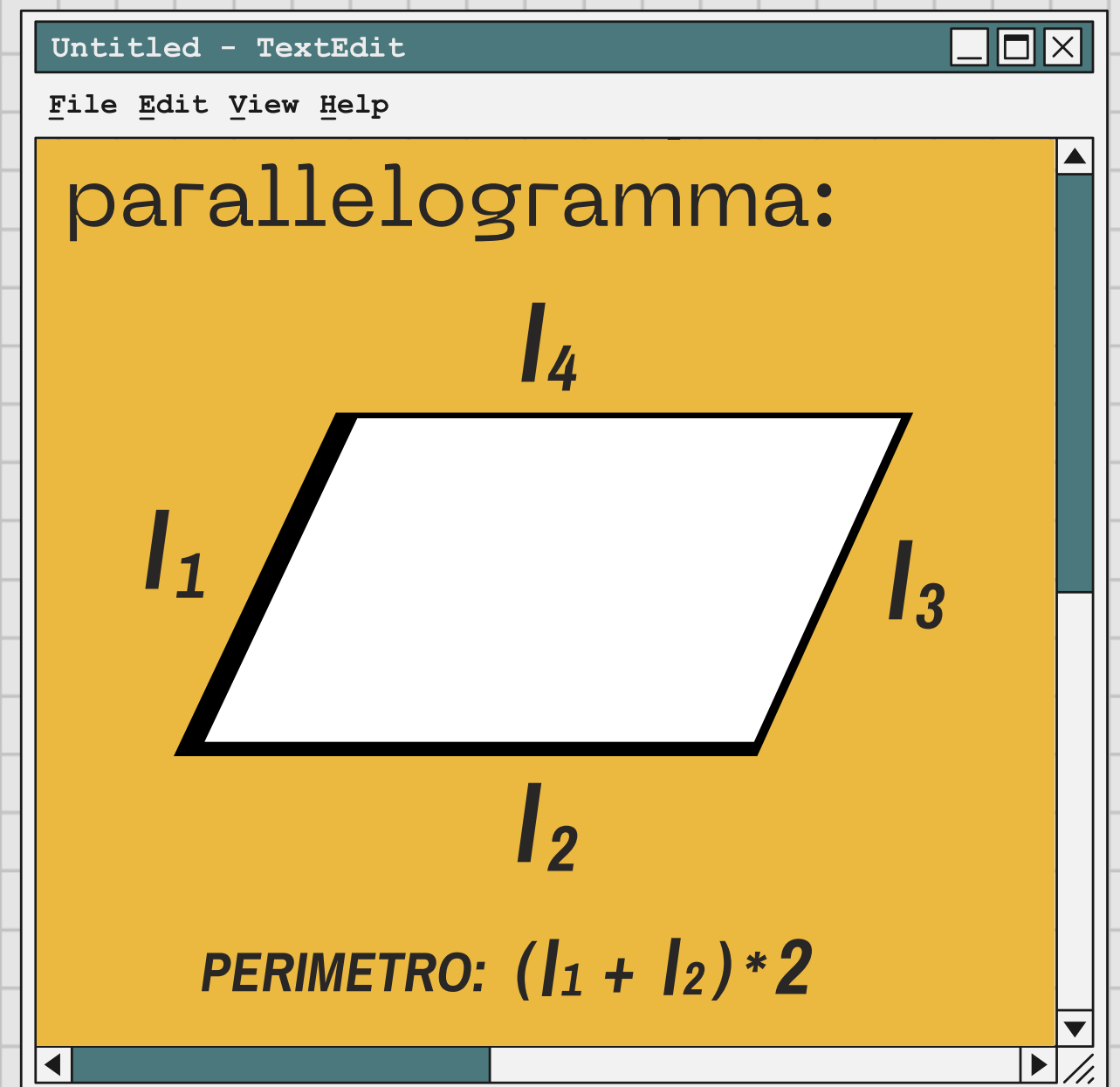
codice Python:

```
def calcola_perimetro_parallelogramma(base, lato):  
    return (base + lato) * 2
```

```
if scelta == '1':  
    base = float(input("Inserisci la base del parallelogramma: "))  
    lato = float(input("Inserisci il lato del parallelogramma: "))  
    print("Perimetro del parallelogramma:", calcola_perimetro_parallelogramma(base, lato))
```

```
$ python3 exercise.py  
Calcolatore di perimetro e circonferenza  
1. Parallelogramma  
2. Cerchio  
3. Triangolo scaleno  
Scegli una figura (1-3): 1  
Inserisci la base del parallelogramma: 100  
Inserisci il lato del parallelogramma: 50  
Perimetro del parallelogramma: 300.0
```

# Figura scelta:





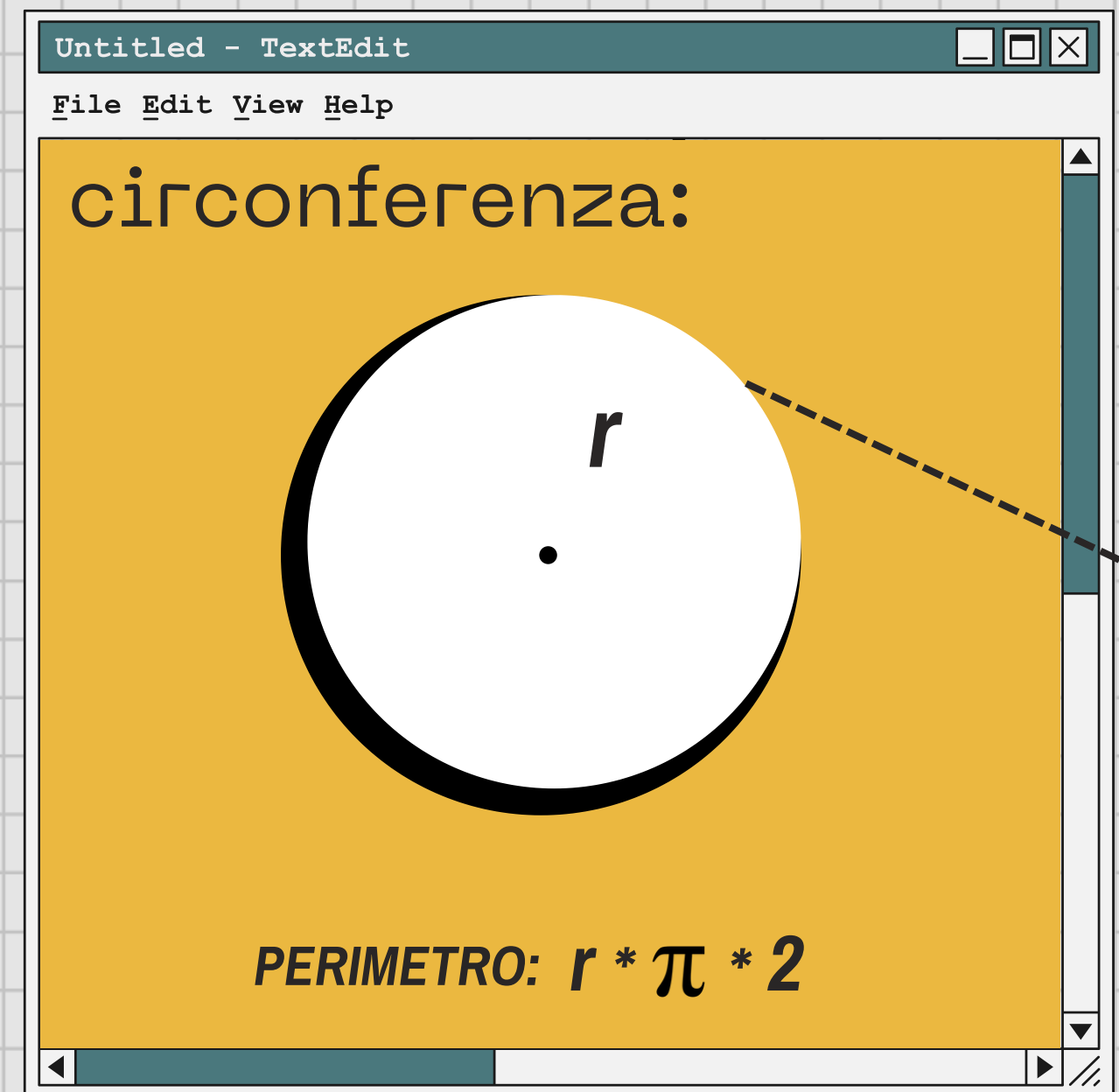
codice Python:

```
def calcola_circonferenza_cerchio(raggio):  
    return 2 * math.pi * raggio
```

```
elif scelta == '2':  
    raggio = float(input("Inserisci il raggio del cerchio: "))  
    print("Circonferenza del cerchio:", calcola_circonferenza_cerchio(raggio))
```

```
(kali@kali)-[~]  
└─$ python3 exercise.py  
Calcolatore di perimetro e circonferenza  
1. Parallelogramma  
2. Cerchio  
3. Triangolo scaleno  
Scegli una figura (1-3): 2  
Inserisci il raggio del cerchio: 100  
Circonferenza del cerchio: 628.3185307179587
```

# Figura scelta:





# CONCLUSIONI

```
GNU nano 7.2
import math

def calcola_perimetro_parallelogramma(base, altezza):
    return (base + altezza) * 2

def calcola_circonferenza_cerchio(raggio):
    return 2 * math.pi * raggio

def calcola_perimetro_triangolo_scaleno(lato_a, lato_b, lato_c):
    return lato_a + lato_b + lato_c

# Funzione principale del programma
def main():
    print("Calcolatore di perimetro e circonferenza")
    print("1. Parallelogramma")
    print("2. Cerchio")
    print("3. Triangolo scaleno") # Modificato da 'Triangolo' a 'Triangolo scaleno'
    scelta = input("Scegli una figura (1-3): ")

    if scelta == '1':
        base = float(input("Inserisci la base del parallelogramma: "))
        altezza = float(input("Inserisci l'altezza del parallelogramma: "))
        print("Perimetro del parallelogramma:", calcola_perimetro_parallelogramma(base, altezza))

    elif scelta == '2':
        raggio = float(input("Inserisci il raggio del cerchio: "))
        print("Circonferenza del cerchio:", calcola_circonferenza_cerchio(raggio))

    elif scelta == '3':
        lato_a = float(input("Inserisci il primo lato del triangolo scaleno: "))
        lato_b = float(input("Inserisci il secondo lato del triangolo scaleno: "))
        lato_c = float(input("Inserisci il terzo lato del triangolo scaleno: "))
        print("Perimetro del triangolo scaleno:", calcola_perimetro_triangolo_scaleno(lato_a, lato_b, lato_c))

    else:
        print("Scelta non valida.")

# Esegui il programma
if __name__ == "__main__":
    main()
```



File Actions Edit View Help

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

```
(kali@kali)-[~]
$ python3 exercise.py
Calcolatore di perimetro e circonferenza
1. Parallelogramma
2. Cerchio
3. Triangolo scaleno
Scegli una figura (1-3): 1
Inserisci la base del parallelogramma: 10
Inserisci l'altezza del parallelogramma: 20
Perimetro del parallelogramma: 60.0
```

```
(kali@kali)-[~]
$ python3 exercise.py
Calcolatore di perimetro e circonferenza
1. Parallelogramma
2. Cerchio
3. Triangolo scaleno
Scegli una figura (1-3): 2
Inserisci il raggio del cerchio: 100
Circonferenza del cerchio: 628.3185307179587
```

```
(kali@kali)-[~]
$ python3 exercise.py
Calcolatore di perimetro e circonferenza
1. Parallelogramma
2. Cerchio
3. Triangolo scaleno
Scegli una figura (1-3): 3
Inserisci il primo lato del triangolo scaleno: 10
Inserisci il secondo lato del triangolo scaleno: 23
Inserisci il terzo lato del triangolo scaleno: 21
Perimetro del triangolo scaleno: 54.0
```

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





# CONSIDERAZIONI FINALI

In fine possiamo evidenziare alcuni punti chiave:

**1) Funzionalità del programma:** Il programma offre una soluzione pratica per calcolare il perimetro di tre figure geometriche comuni: il parallelogramma, il cerchio e il triangolo. Ogni funzione è progettata per calcolare il perimetro specifico della figura geometrica corrispondente.

**2) Interazione con l'utente:** L'interfaccia utente è progettata per essere intuitiva e facile da usare. L'utente può selezionare la figura geometrica desiderata e inserire i valori necessari per ottenere il risultato del calcolo.

**3) Modularità del codice:** Il codice è organizzato in modo modulare, con funzioni separate per calcolare il perimetro di ciascuna figura geometrica, questo rende il programma più leggibile e facilita eventuali future estensioni o modifiche.



**4) Gestione degli errori:** Il programma gestisce i casi in cui l'utente inserisce input non validi o valori negativi per le misure delle figure geometriche, garantendo una corretta esecuzione del calcolo del perimetro.

In conclusione, il programma offre un'implementazione efficace e pratica per il calcolo del perimetro di diverse figure geometriche, fornendo un'interfaccia user-friendly e una gestione accurata degli input dell'utente e degli errori. La sua modularità lo rende adattabile a ulteriori esigenze o sviluppi futuri

