

Calcolatore di perimetri

apro editor nano per scrivere il mio programma

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

Traccia: Si scriva un programma in Python che in base alla scelta dell'utente permetta di calcolare il perimetro di:

- Quadrato (perimetro = lato*4)
- Cerchio (circonferenza = $2 \cdot \pi \cdot \text{raggio}$)
- Rettangolo (perimetro = base*2 + altezza*2)

Come si presenta il programma scritto una volta eseguito + verifica dei controlli inseriti:
per eseguire

```
(kali@kali)-[~]  
$ python3 perimetri.py
```

Calcola il perimetro di una delle seguenti figure geometriche

1. Quadrato
2. Cerchio
3. Rettangolo
4. Esci

Scegli un'opzione (1-4): 1

Inserisci il lato del quadrato: 3

Perimetro del quadrato: 12.0

Calcola il perimetro di una delle seguenti figure geometriche

1. Quadrato
2. Cerchio
3. Rettangolo
4. Esci

Scegli un'opzione (1-4): 2

Inserisci il raggio del cerchio: 1

Circonferenza del cerchio: 6.283185307179586

Calcola il perimetro di una delle seguenti figure geometriche

1. Quadrato
2. Cerchio
3. Rettangolo
4. Esci

Scegli un'opzione (1-4): 3

Inserisci la base del rettangolo: 3

Inserisci l'altezza del rettangolo: 5

Perimetro del rettangolo: 16.0

Calcola il perimetro di una delle seguenti figure geometriche

1. Quadrato
2. Cerchio
3. Rettangolo
4. Esci

Scegli un'opzione (1-4): a

Scelta non valida. Riprova.

Calcola il perimetro di una delle seguenti figure geometriche

1. Quadrato
2. Cerchio
3. Rettangolo
4. Esci

Scegli un'opzione (1-4): 12

Scelta non valida. Riprova.

Calcola il perimetro di una delle seguenti figure geometriche

1. Quadrato
2. Cerchio
3. Rettangolo
4. Esci

Scegli un'opzione (1-4): 4

Fine del programma.

Codice del programma:

```
GNU nano 7.2                                perimetri.py *
# Importo il modulo math per accedere a funzioni matematiche come pi
import math
# Definisco una funzione per calcolare il perimetro di un quadrato
def perimetro_quadrato(lato):
    # Moltiplico il lato per 4
    return lato * 4
# Definisco una funzione per calcolare la circonferenza di un cerchio
def perimetro_cerchio(raggio):
    # Uso la formula 2πr
    return 2 * math.pi * raggio
# Definisco una funzione per calcolare il perimetro di un rettangolo
def perimetro Rettangolo(base, altezza):
    # Sommo base e altezza e moltiplico per 2
    return 2 * (base + altezza)

# Definisco una funzione ausiliaria per gestire l'input numerico
def input_numero(prompt):
    # Inizio un loop infinito per richiedere l'input all'utente
    while True:
        try:
            # Provo a convertire l'input in un float
            return float(input(prompt))
            # Se la conversione fallisce, catturo l'eccezione
        except ValueError:
            # Informo l'utente dell'errore
            print("Inserisci un numero valido.")

# Definisco la funzione principale del programma
def main():
    # Inizio un altro loop infinito per il menù principale
    while True:
        # Stampo il menù delle opzioni
        print("\nCalcola il perimetro di una delle seguenti figure geometriche")
        print("1. Quadrato")
        print("2. Cerchio")
        print("3. Rettangolo")
        print("4. Esci")
        # Chiedo all'utente di scegliere un'opzione
        scelta = input("Scegli un'opzione (1-4): ")

        if scelta == '1':
            lato = input_numero("Inserisci il lato del quadrato: ")
            print("Perimetro del quadrato:", perimetro_quadrato(lato))
        elif scelta == '2':
            raggio = input_numero("Inserisci il raggio del cerchio: ")
            print("Circonferenza del cerchio:", perimetro_cerchio(raggio))
        elif scelta == '3':
            base = input_numero("Inserisci la base del rettangolo: ")
            altezza = input_numero("Inserisci l'altezza del rettangolo: ")
            print("Perimetro del rettangolo:", perimetro_Rettangolo(base, altezza))
        elif scelta == '4':
            # Se l'utente sceglie di uscire, termino il loop
            print("Fine del programma.")
            break
        else:
            # Se l'opzione non è valida, lo comunico all'utente
            print("Scelta non valida. Riprova.")

if __name__ == "__main__":
    # Eseguo la funzione main se il programma è eseguito direttamente
    main()
```