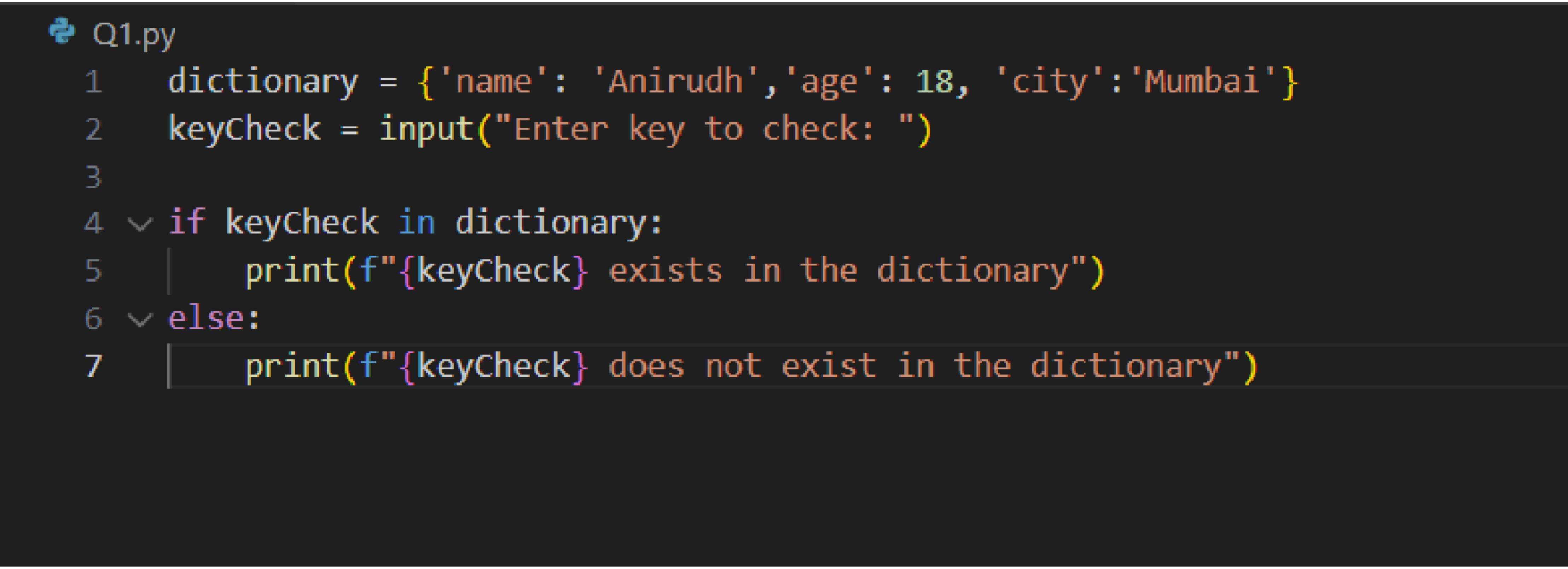
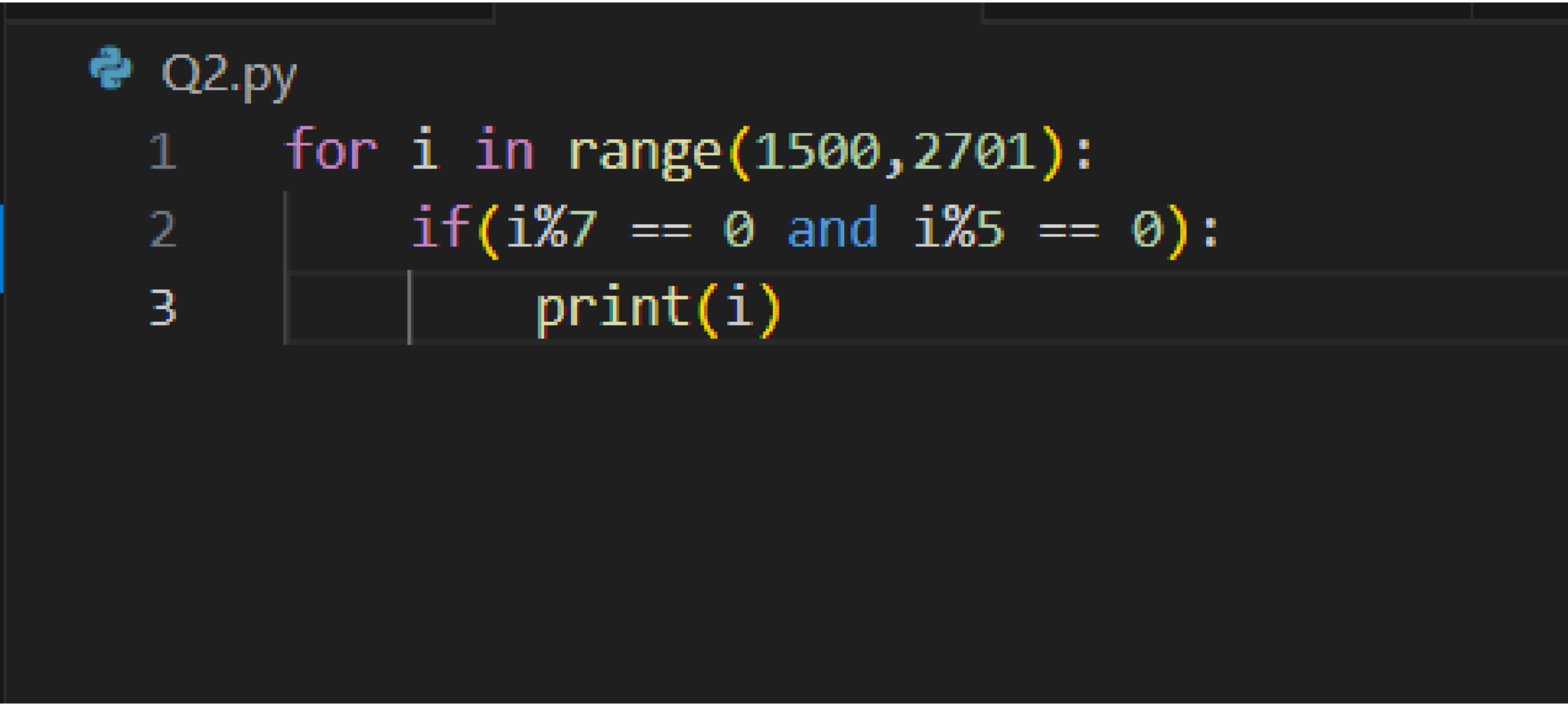
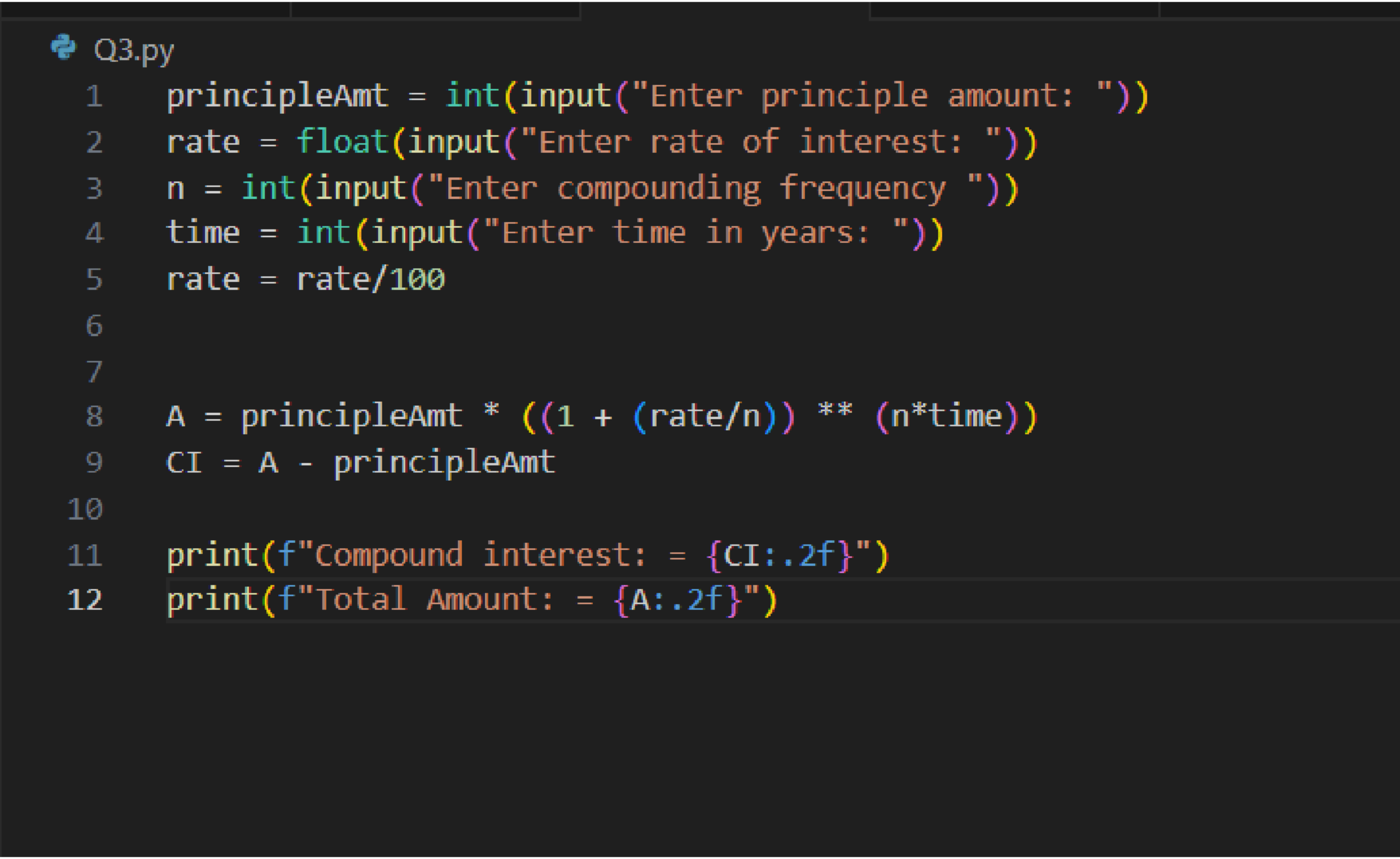


```python
dictionary = {'name': 'Anirudh','age': 18, 'city':'Mumbai'}
keyCheck = input("Enter key to check: ")

if keyCheck in dictionary:
    print(f"{keyCheck} exists in the dictionary")
else:
    print(f"{keyCheck} does not exist in the dictionary")
```

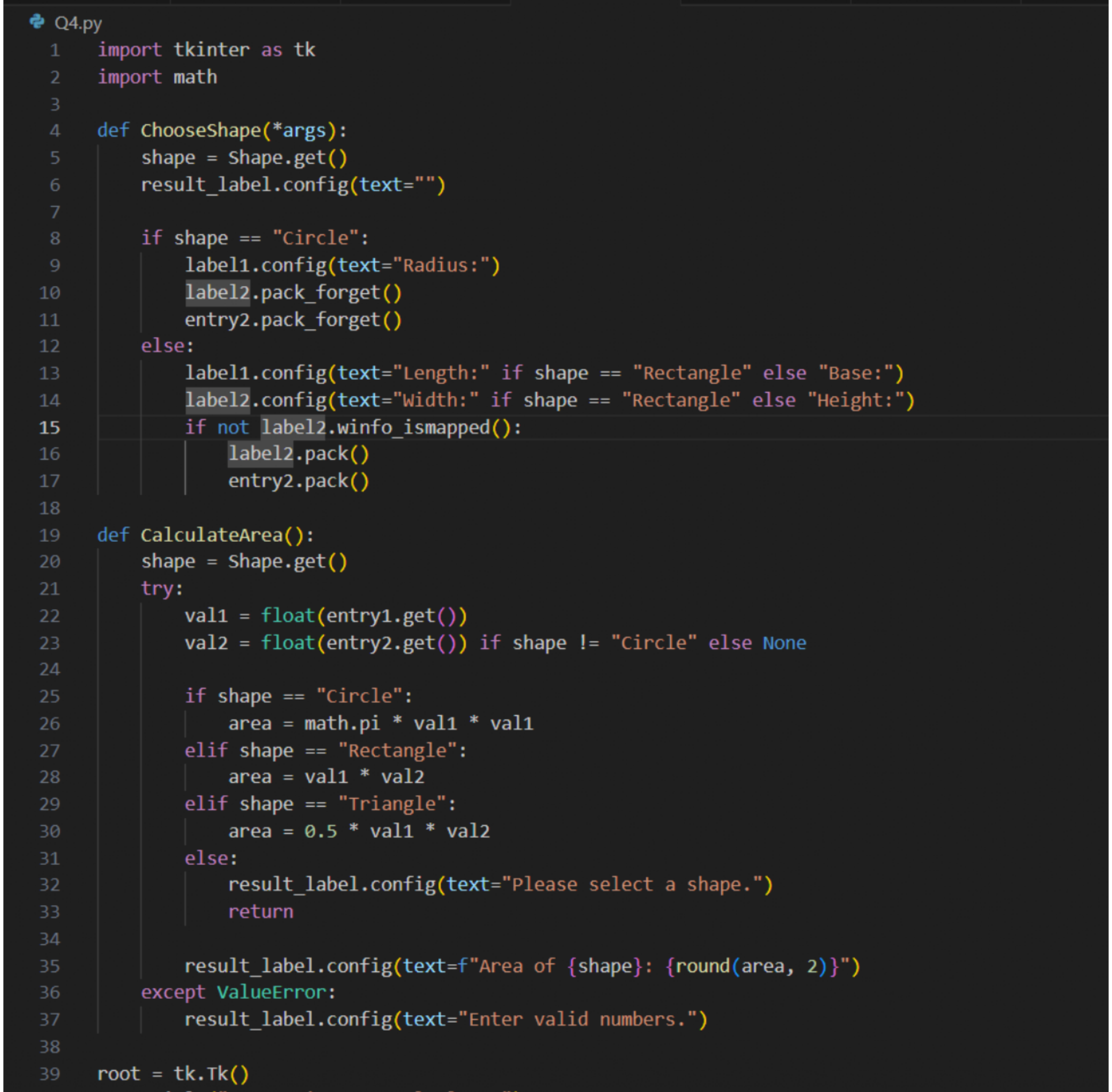


```python
for i in range(1500,2701):
    if(i%7 == 0 and i%5 == 0):
        print(i)
```

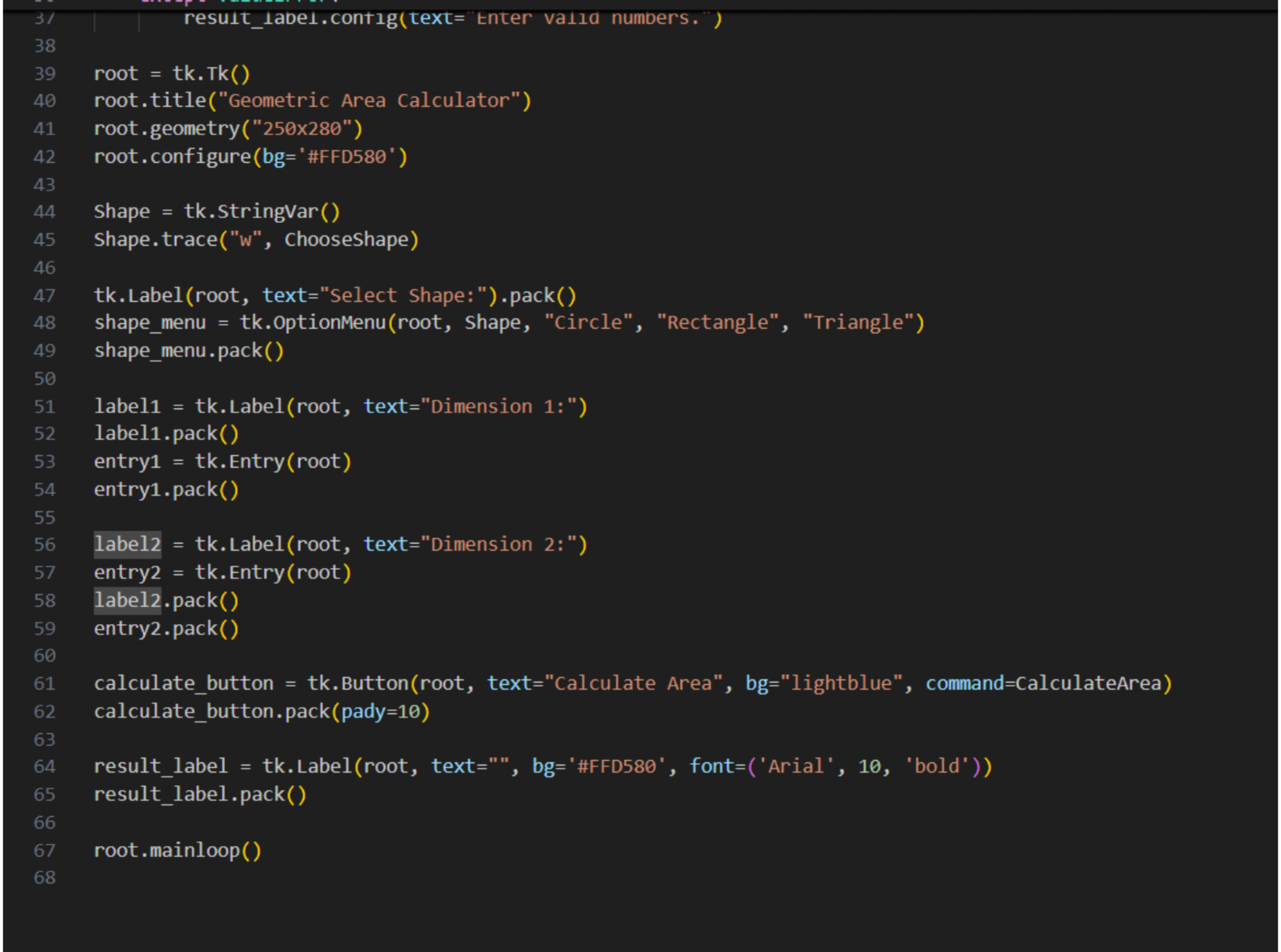


Q3.py

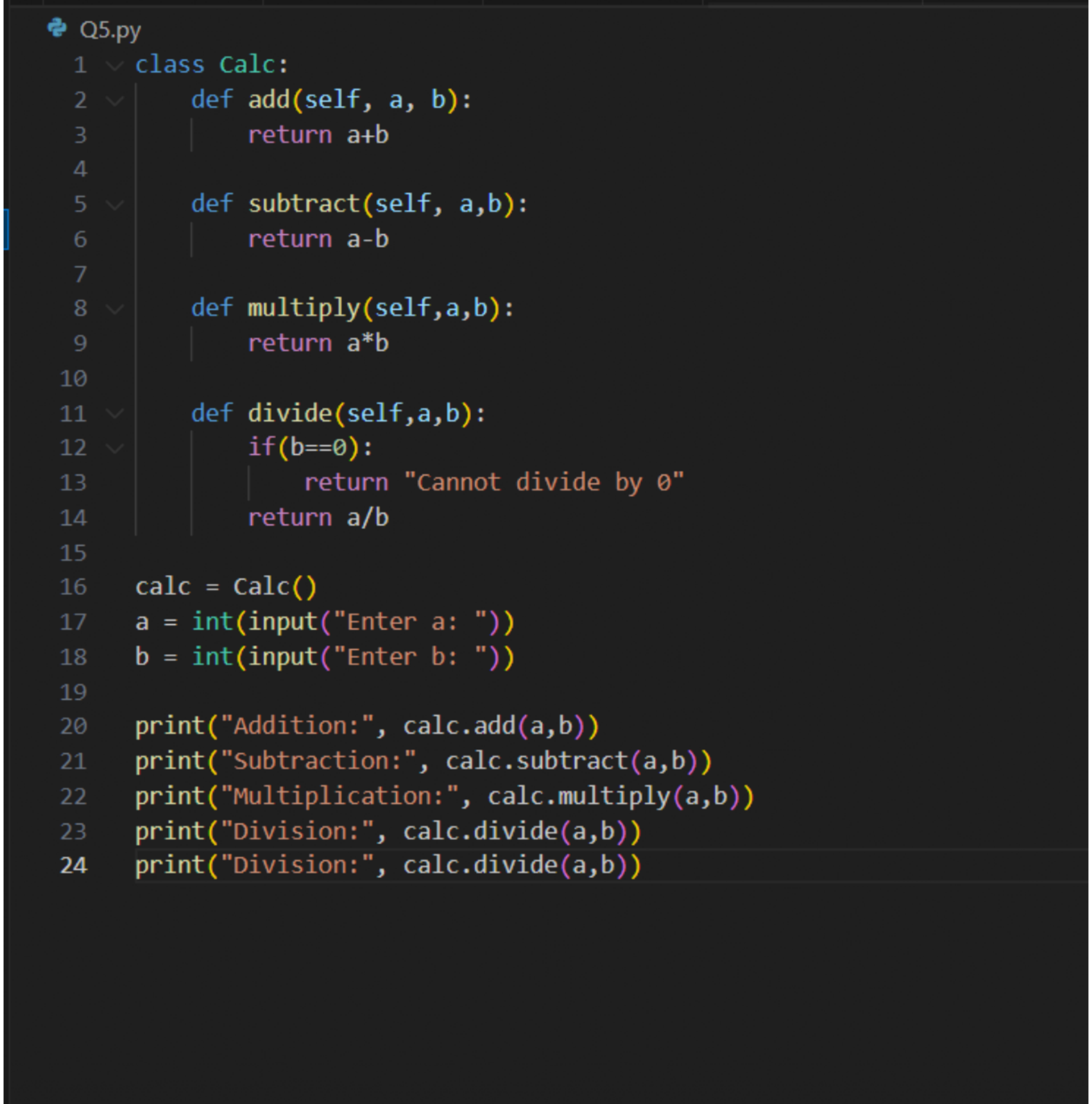
```python
principleAmt = int(input("Enter principle amount: "))
rate = float(input("Enter rate of interest: "))
n = int(input("Enter compounding frequency "))
time = int(input("Enter time in years: "))
rate = rate/100


A = principleAmt * ((1 + (rate/n)) ** (n*time))
CI = A - principleAmt

print(f"Compound interest: = {CI:.2f}")
print(f"Total Amount: = {A:.2f}")
```



```python
import tkinter as tk
import math

def ChooseShape(*args):
    shape = Shape.get()
    result_label.config(text="")

    if shape == "Circle":
        label1.config(text="Radius:")
        label2.pack_forget()
        entry2.pack_forget()
    else:
        label1.config(text="Length:" if shape == "Rectangle" else "Base:")
        label2.config(text="Width:" if shape == "Rectangle" else "Height:")
        if not label2.winfo_ismapped():
            label2.pack()
            entry2.pack()

def CalculateArea():
    shape = Shape.get()
    try:
        val1 = float(entry1.get())
        val2 = float(entry2.get()) if shape != "Circle" else None

        if shape == "Circle":
            area = math.pi * val1 * val1
        elif shape == "Rectangle":
            area = val1 * val2
        elif shape == "Triangle":
            area = 0.5 * val1 * val2
        else:
            result_label.config(text="Please select a shape.")
            return

        result_label.config(text=f"Area of {shape}: {round(area, 2)}")
    except ValueError:
        result_label.config(text="Enter valid numbers.")

root = tk.Tk()
```

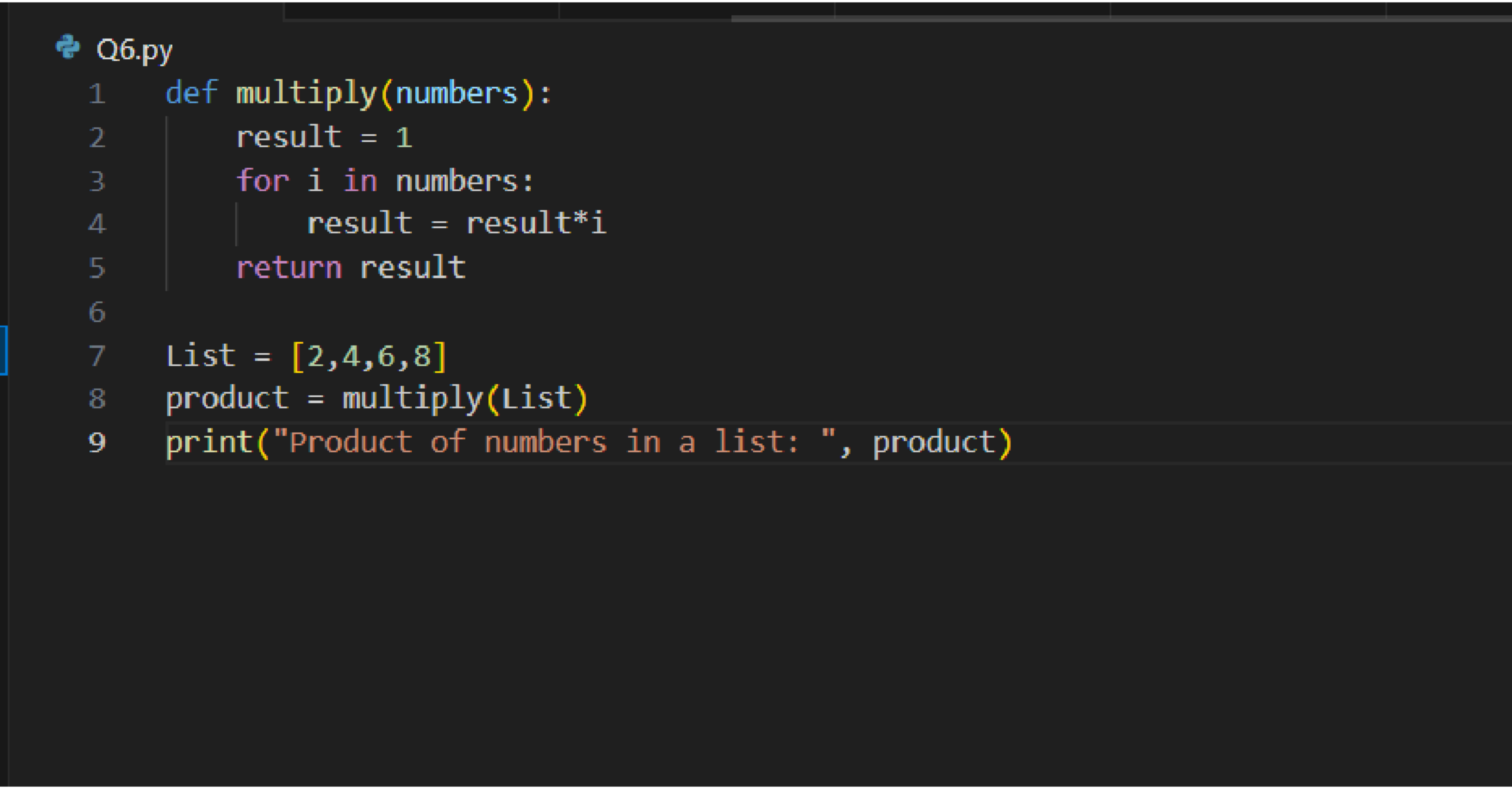


```
        result_label.config(text="Enter valid numbers.")

root = tk.Tk()
root.title("Geometric Area Calculator")
root.geometry("250x280")
root.configure(bg='#FFD580')

Shape = tk.StringVar()
Shape.trace("w", ChooseShape)

tk.Label(root, text="Select Shape:").pack()
shape_menu = tk.OptionMenu(root, Shape, "Circle", "Rectangle", "Triangle")
shape_menu.pack()

label1 = tk.Label(root, text="Dimension 1:")
label1.pack()
entry1 = tk.Entry(root)
entry1.pack()

label2 = tk.Label(root, text="Dimension 2:")
entry2 = tk.Entry(root)
label2.pack()
entry2.pack()

calculate_button = tk.Button(root, text="Calculate Area", bg="lightblue", command=CalculateArea)
calculate_button.pack(pady=10)

result_label = tk.Label(root, text="", bg='#FFD580', font=('Arial', 10, 'bold'))
result_label.pack()

root.mainloop()

```

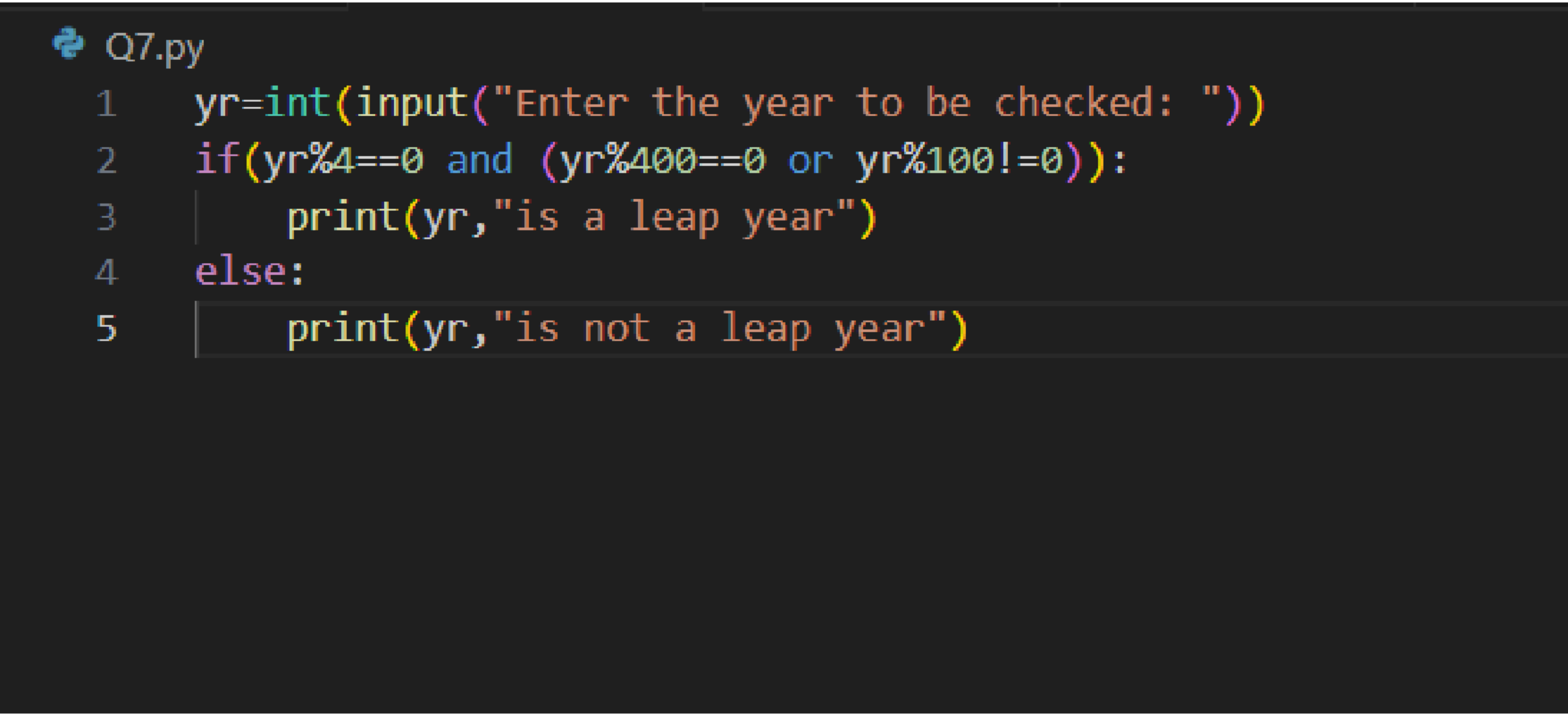


Q5.py

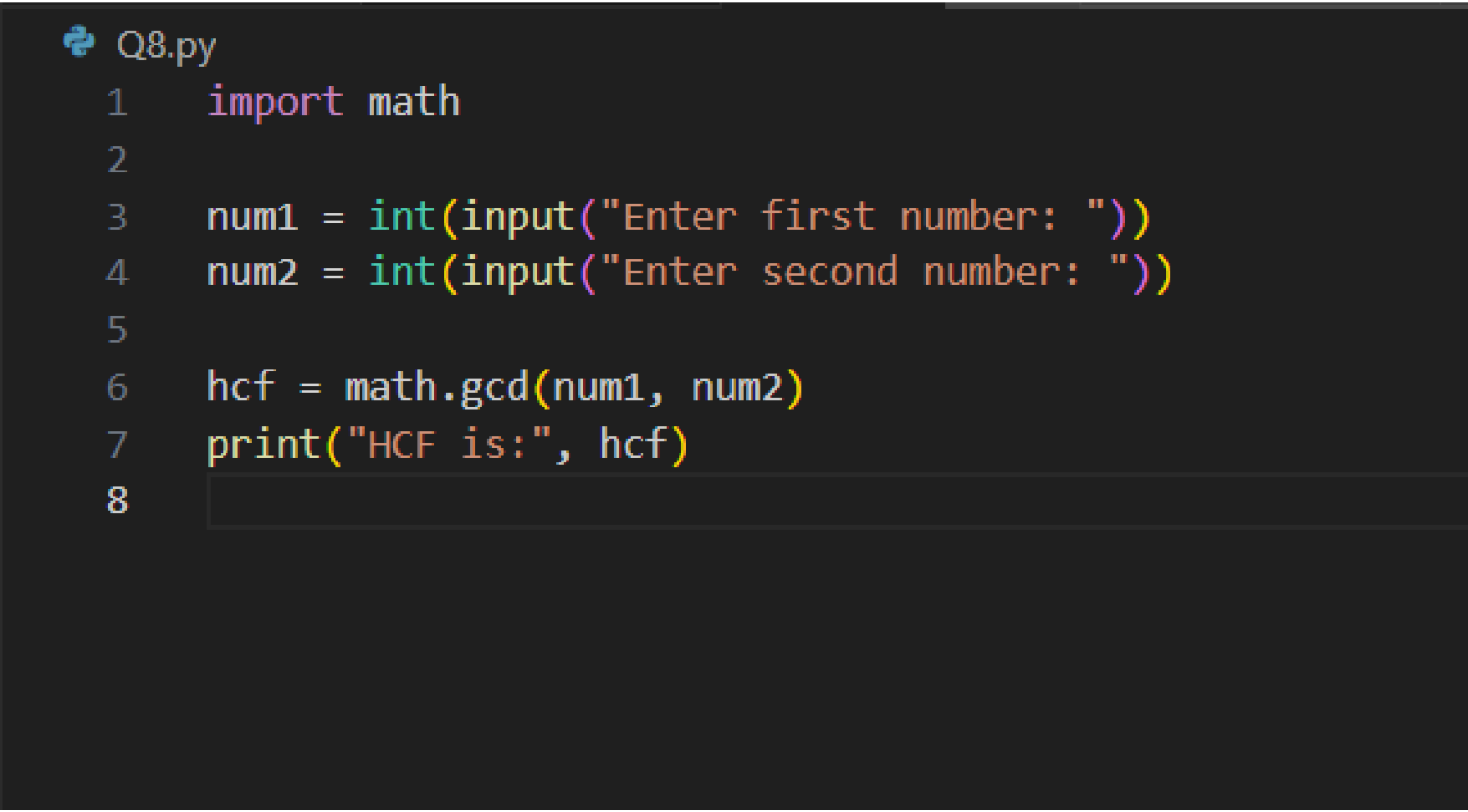
```python
class Calc:
    def add(self, a, b):
        return a+b

    def subtract(self, a,b):
        return a-b

    def multiply(self,a,b):
        return a*b

    def divide(self,a,b):
        if(b==0):
            return "Cannot divide by 0"
        return a/b

calc = Calc()
a = int(input("Enter a: "))
b = int(input("Enter b: "))

print("Addition:", calc.add(a,b))
print("Subtraction:", calc.subtract(a,b))
print("Multiplication:", calc.multiply(a,b))
print("Division:", calc.divide(a,b))
print("Division:", calc.divide(a,b))
```



```python
def multiply(numbers):
    result = 1
    for i in numbers:
        result = result*i
    return result

List = [2,4,6,8]
product = multiply(List)
print("Product of numbers in a list: ", product)
```

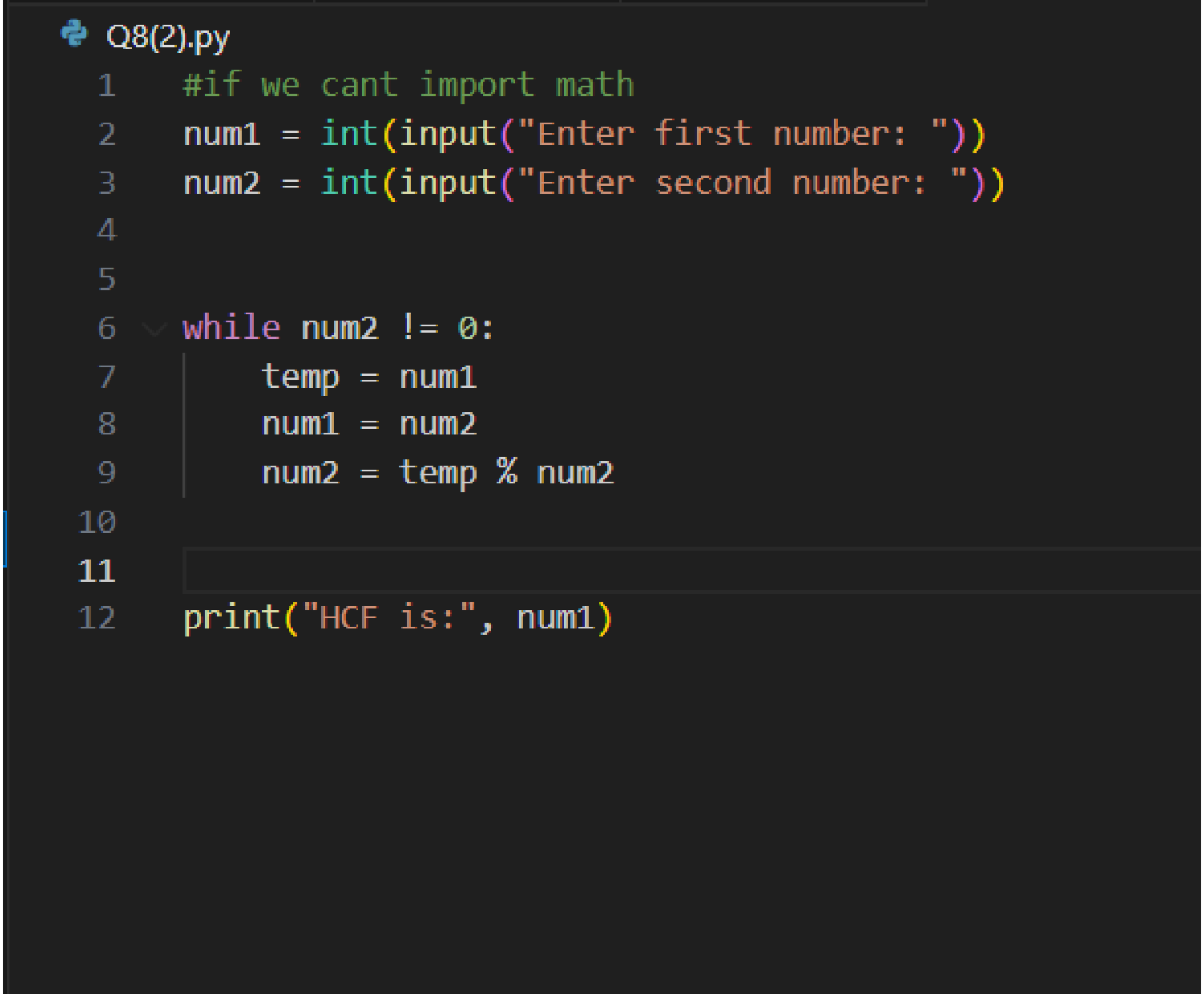


```python
yr=int(input("Enter the year to be checked: "))
if(yr%4==0 and (yr%400==0 or yr%100!=0)):
    print(yr,"is a leap year")
else:
    print(yr,"is not a leap year")
```

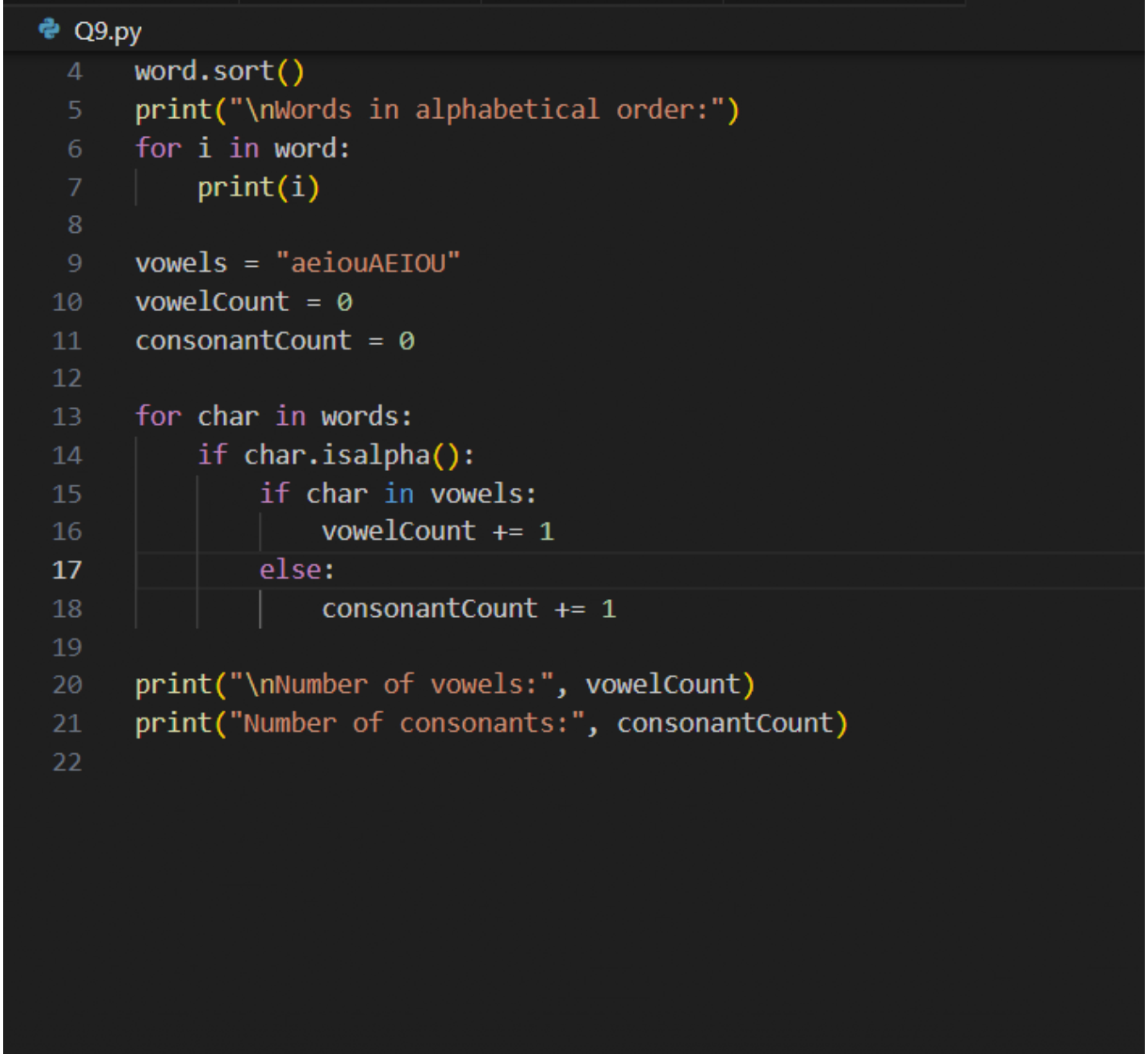


Q8.py

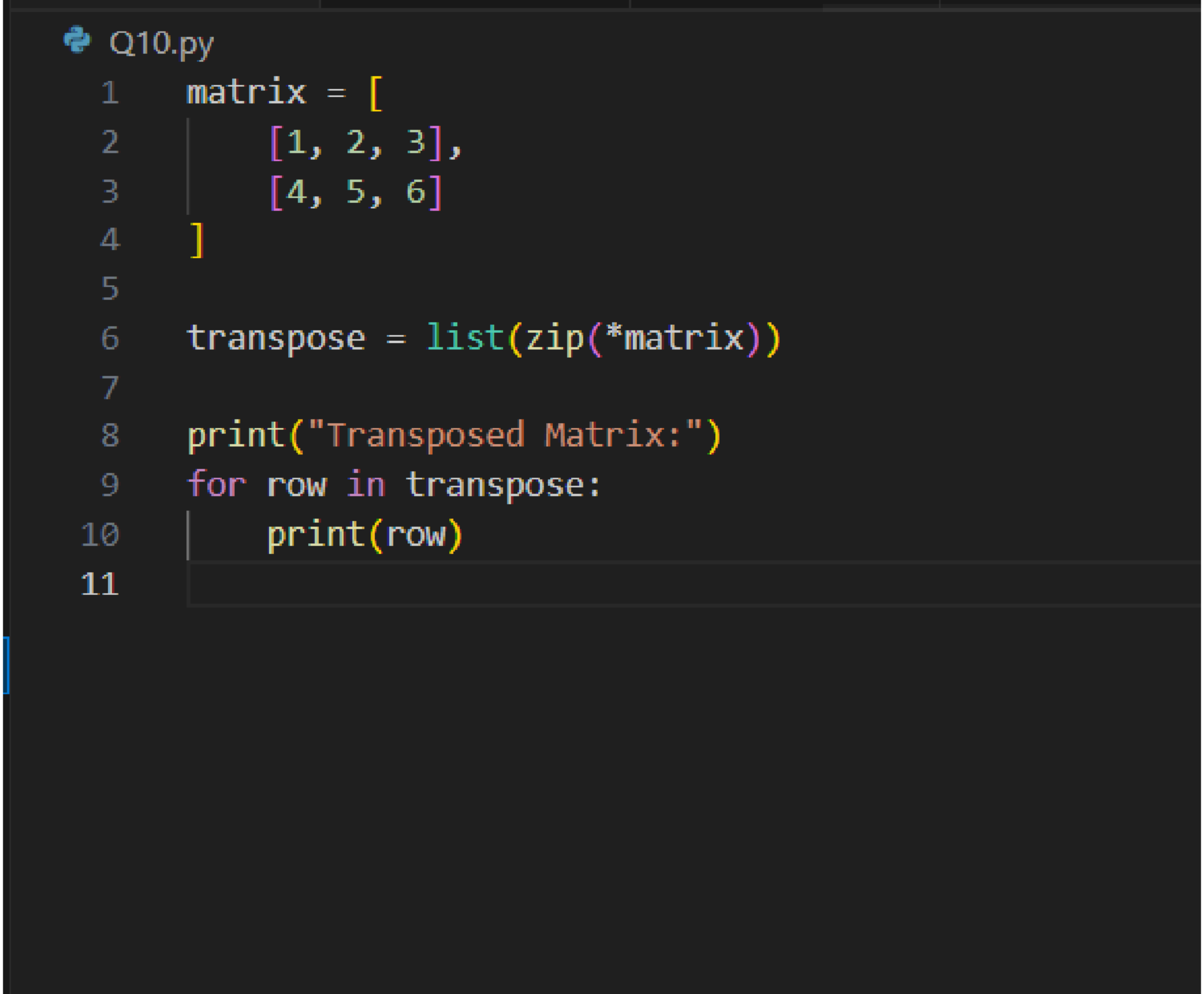
```python
import math

num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))

hcf = math.gcd(num1, num2)
print("HCF is:", hcf)

```



```python
#if we cant import math
num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))


while num2 != 0:
    temp = num1
    num1 = num2
    num2 = temp % num2


print("HCF is:", num1)
```

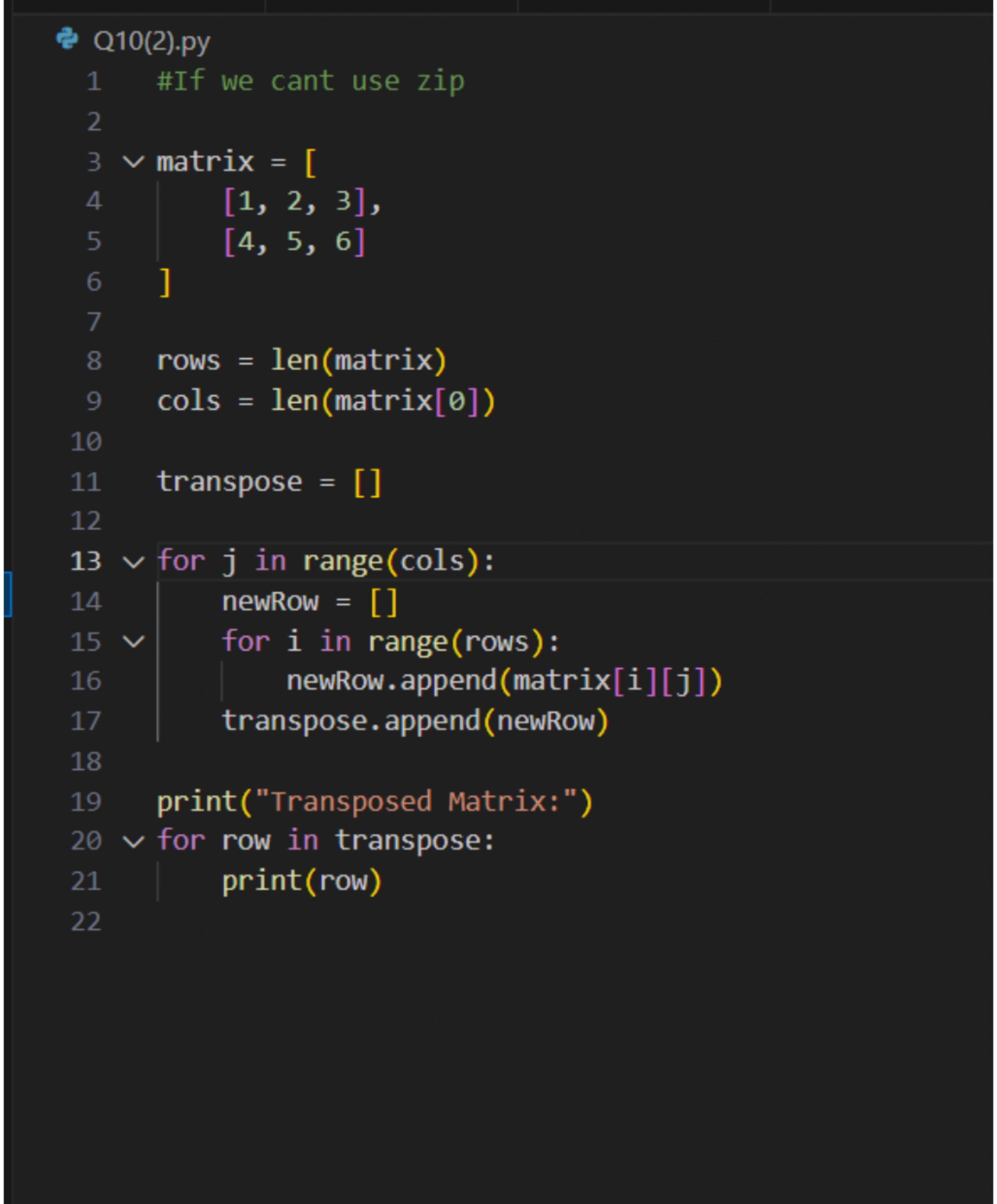


```python
word.sort()
print("\nWords in alphabetical order:")
for i in word:
    print(i)

vowels = "aeiouAEIOU"
vowelCount = 0
consonantCount = 0

for char in words:
    if char.isalpha():
        if char in vowels:
            vowelCount += 1
        else:
            consonantCount += 1

print("\nNumber of vowels:", vowelCount)
print("Number of consonants:", consonantCount)
```



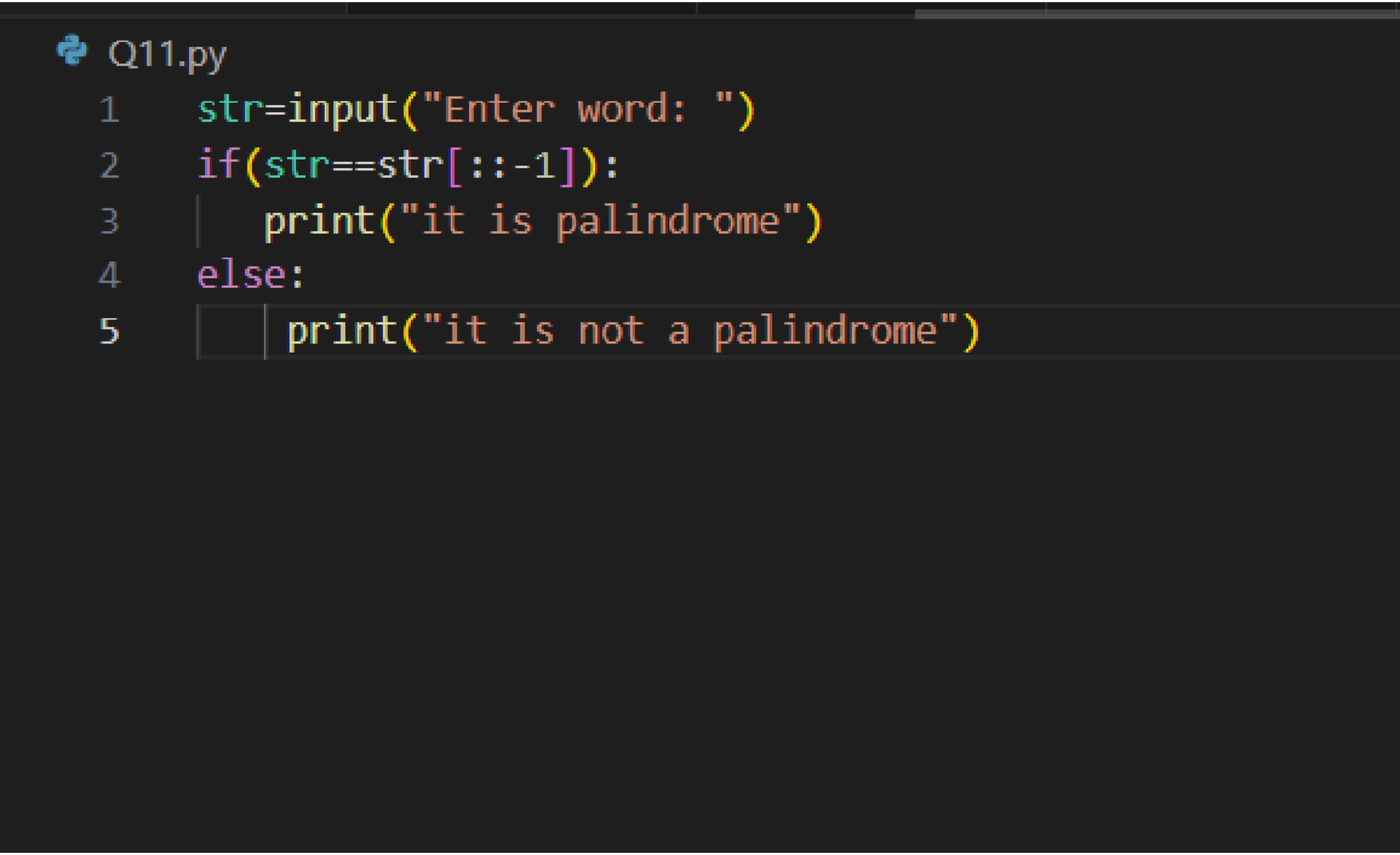
Q10.py

```python
matrix = [
    [1, 2, 3],
    [4, 5, 6]
]

transpose = list(zip(*matrix))

print("Transposed Matrix:")
for row in transpose:
    print(row)

```



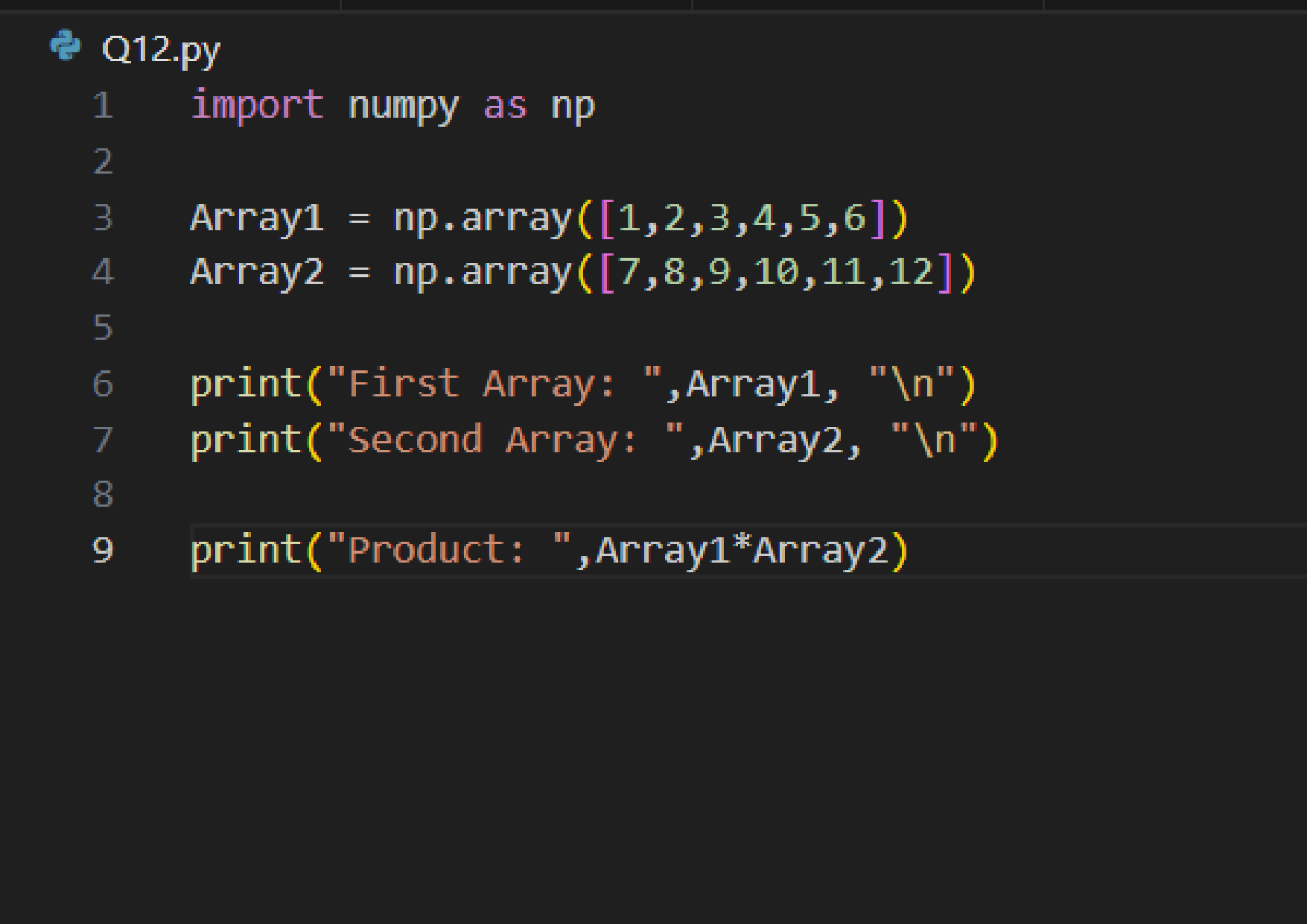
Q10(2).py

```python
#If we cant use zip

matrix = [
        [1, 2, 3],
        [4, 5, 6]
]

rows = len(matrix)
cols = len(matrix[0])

transpose = []

for j in range(cols):
    newRow = []
    for i in range(rows):
        newRow.append(matrix[i][j])
    transpose.append(newRow)

print("Transposed Matrix:")
for row in transpose:
    print(row)

```



Q11.py

```python
str=input("Enter word: ")
if(str==str[::-1]):
   print("it is palindrome")
else:
    print("it is not a palindrome")
```



```python
import numpy as np

Array1 = np.array([1,2,3,4,5,6])
Array2 = np.array([7,8,9,10,11,12])

print("First Array: ",Array1, "\n")
print("Second Array: ",Array2, "\n")

print("Product: ",Array1*Array2)
```