```python
# -*- coding: utf-8 -*-
"""PPA1.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1JsmtDRHa70MoRBRS4t-Eslp2ihwczFkQ
"""

class Student:
    def __init__(self, name, age, dob, branch, college, email, phone):
        self.name = name
        self.age = age
        self.dob = dob
        self.branch = branch
        self.college = college
        self.email = email
        self.phone = phone

    def display_bio_data(self):
        print("Name:", self.name)
        print("Age:", self.age)
        print("Date of Birth:", self.dob)
        print("Branch:", self.branch)
        print("College:", self.college)
        print("Email ID:", self.email)
        print("Phone Number:", self.phone)


# Example usage:
if __name__ == "__main__":
    # Creating a Student object
    student1 = Student("John Doe", 20, "2004-05-15", "Computer Science", "ABC College", "john.doe@example.com", "1234567890")

    # Displaying the bio data of the student
    student1.display_bio_data()

# Initialize an empty list to store the numbers
result = []

# Iterate through numbers from 1500 to 2700 (inclusive)
for num in range(1500, 2701):
    # Check if the number is divisible by 7 and a multiple of 5
    if num % 7 == 0 and num % 5 == 0:
        # If true, append the number to the result list
        result.append(num)

# Print the numbers
print("Numbers between 1500 and 2700 that are divisible by 7 and multiples of 5:")
print(result)

def calculate_total_and_average(marks):
    total_marks = sum(marks)
    average_marks = total_marks / len(marks)
    return total_marks, average_marks

def main():
    # Initialize an empty list to store the marks
    marks = []

    # Input marks for 5 subjects
    for i in range(1, 6):
        subject_mark = float(input(f"Enter the marks for subject {i}: "))
        marks.append(subject_mark)

    # Calculate total and average marks
    total_marks,

def calculator(operator, num1, num2):
    if operator == "+":
        result = num1 + num2
    elif operator == "-":
        result = num1 - num2
    elif operator == "*":
        result = num1 * num2
    elif operator == "/":
        if num2 != 0:
            result = num1 / num2
        else:
            result = "Error: Division by zero!"
    else:
        result = "Error: Invalid operator!"
```

```
    return result

# Example usage:
operator = input("Enter the operator (+, -, *, /): ")
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))

result = calculator(operator, num1, num2)
print("Result:", result)

def triangle_type(side1, side2, side3):
    if side1 == side2 == side3:
        return "Equilateral"
    elif side1 == side2 or side1 == side3 or side2 == side3:
        return "Isosceles"
    else:
        return "Scalene"

def main():
    side1 = float(input("Enter the length of side 1: "))
    side2 = float(input("Enter the length of side 2: "))
    side3 = float(input("Enter the length of side 3: "))

    triangle = triangle_type(side1, side2, side3)
    print("The triangle is:", triangle)

if __name__ == "__main__":
    main()

def swap_with_temp(a, b):
    temp = a
    a = b
    b = temp
    return a, b

# Example usage:
num1 = 5
num2 = 10
print("Before swapping:", num1, num2)
num1, num2 = swap_with_temp(num1, num2)
print("After swapping:", num1, num2)

def swap_without_temp(a, b):
    a = a + b
    b = a - b
    a = a - b
    return a, b

# Example usage:
num1 = 5
num2 = 10
print("Before swapping:", num1, num2)
num1, num2 = swap_without_temp(num1, num2)
print("After swapping:", num1, num2)

def fizz_buzz(n):
    for i in range(1, n + 1):
        if i % 3 == 0 and i % 5 == 0:
            print("FizzBuzz")
        elif i % 3 == 0:
            print("Fizz")
        elif i % 5 == 0:
            print("Buzz")
        else:
            print(str(i))

# Example usage:
n = int(input("Enter the value of n: "))
fizz_buzz(n)

import math

def calculate_area(a, b, c):
    # Calculate semi-perimeter
    s = (a + b + c) / 2

    # Calculate area using Heron's formula
    area = math.sqrt(s * (s - a) * (s - b) * (s - c))
    return area

# Example usage:
```

```
side1 = float(input("Enter the length of side 1: "))
side2 = float(input("Enter the length of side 2: "))
side3 = float(input("Enter the length of side 3: "))

area = calculate_area(side1, side2, side3)
print("Area of the triangle:", area)

import math

def square_and_square_root(number):
    square = number ** 2
    square_root = math.sqrt(number)
    return square, square_root

# Example usage:
number = float(input("Enter a number: "))

square, square_root = square_and_square_root(number)
print("Square of", number, ":", square)
print("Square root of", number, ":", square_root)

import math

def classify_triangle(a, b, c):
    # Calculate the squares of the side lengths
    a_squared = a ** 2
    b_squared = b ** 2
    c_squared = c ** 2

    # Calculate the cosine of each angle using the Law of Cosines
    cos_a = (b_squared + c_squared - a_squared) / (2 * b * c)

def find_greatest(num1, num2, num3):
    if num1 >= num2 and num1 >= num3:
        return num1
    elif num2 >= num1 and num2 >= num3:
        return num2
    else:
        return num3

# Example usage:
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
num3 = float(input("Enter the third number: "))

greatest = find_greatest(num1, num2, num3)
print("The greatest number is:", greatest)

def calculate_grade(marks):
    if marks >= 90:
        return 'A+'
    elif marks >= 80:
        return 'A'
    elif marks >= 70:
        return 'B'
    elif marks >= 60:
        return 'C'
    elif marks >= 50:
        return 'D'
    else:
        return 'F'

# Example usage:
marks = float(input("Enter the marks: "))

grade = calculate_grade(marks)
print("Grade:", grade)

def is_leap_year(year):
    if year % 4 == 0:
        if year % 100 == 0:
            if year % 400 == 0:
                return True  # Divisible by 400 is a leap year
            else:
                return False
        else:
            return True  # Divisible by 4 but not by 100 is a leap year
    else:
        return False  # Not divisible by 4 is not a leap year

# Example usage:
year = int(input("Enter a year: "))
```

```python
if is_leap_year(year):
    print(year, "is a leap year.")
else:
    print(year, "is not a leap year.")

import random

def guess_number():
    # Generate a random number between 1 and 9
    secret_number = random.randint(1, 9)

    # Prompt the user to guess the number
    while True:
        guess = int(input("Guess a number between 1 and 9: "))

        # Check if the guess is correct
        if guess == secret_number:
            print("Congratulations! You guessed the number", secret_number, "correctly!")
            break
        else:
            print("Wrong guess. Try again!")

# Run the game
guess_number()

# Using list comprehension
squared_numbers = [x ** 2 for x in range(1, 21)]

# Print the list
print("List of squares of numbers between 1 and 20:")
print(squared_numbers)

def sum_of_digits(number):
    # Initialize sum to 0
    total = 0

    # Iterate through each digit of the number
    while number > 0:
        # Extract the last digit using modulus operator
        digit = number % 10
        # Add the digit to the total
        total += digit
        # Remove the last digit from the number
        number //= 10

    return total

# Example usage:
number = int(input("Enter a number: "))
result = sum_of_digits(number)
print("Sum of the digits of the number:", result)

def is_palindrome(number):
    # Convert the number to a string for easy comparison
    num_str = str(number)

    # Check if the string is equal to its reverse
    return num_str == num_str[::-1]

# Example usage:
number = int(input("Enter a number: "))

if is_palindrome(number):
    print(number, "is a palindrome.")
else:
    print(number, "is not a palindrome.")

def is_divisible(number, divisor):
    # Check if number is divisible by divisor
    return number % divisor == 0

# Example usage:
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number (divisor): "))

if is_divisible(num1, num2):
    print(num1, "is divisible by", num2)
else:
    print(num1, "is not divisible by", num2)

def is_prime(number):
```

```python
    if number <= 1:
        return False  # Numbers less than or equal to 1 are not prime

    # Check for divisibility from 2 to the square root of the number
    for i in range(2, int(number ** 0.5) + 1):
        if number % i == 0:
            return False  # If the number is divisible by any number in this range, it's not prime
    return True  # If the number is not divisible by any number in the range, it's prime

# Example usage:
num = int(input("Enter a number: "))

if is_prime(num):
    print(num, "is a prime number.")
else:
    print(num, "is not a prime number.")

def is_in_range(number, start, end):
    return start <= number <= end

# Example usage:
number = int(input("Enter a number: "))
start_range = int(input("Enter the start of the range: "))
end_range = int(input("Enter the end of the range: "))

if is_in_range(number, start_range, end_range):
    print(number, "is in the range from", start_range, "to", end_range)
else:
    print(number, "is not in the range from", start_range, "to", end_range)

def is_armstrong(number):
    # Count the number of digits in the given number
    num_digits = len(str(number))

    # Calculate the sum of each digit raised to the power of the number of digits
    total = sum(int(digit) ** num_digits for digit in str(number))

    # Check if the total is equal to the original number
    return total == number

# Example usage:
num = int(input("Enter a number: "))

if is_armstrong(num):
    print(num, "is an Armstrong number.")
else:
    print(num, "is not an Armstrong number.")

def is_armstrong(num):
    # Convert the number to a string to determine its length
    num_str = str(num)
    # Calculate the number of digits in the number
    num_digits = len(num_str)
    # Initialize a variable to store the sum of the digits raised to the power of the number of digits
    armstrong_sum = 0
    # Iterate over each digit in the number
    for digit in num_str:
        # Add the digit raised to the power of the number of digits to the sum
        armstrong_sum += int(digit) ** num_digits
    # Check if the sum is equal to the original number
    return armstrong_sum == num

# Example usage:
number = int(input("Enter a number: "))
if is_armstrong(number):
    print(number, "is an Armstrong number.")
else:
    print(number, "is not an Armstrong number.")

def sum_of_divisors(number):
    # Initialize the sum of divisors
    divisor_sum = 0

    # Iterate from 1 to number // 2 (inclusive) to find divisors
    for i in range(1, number // 2 + 1):
        if number % i == 0:
            divisor_sum += i

    return divisor_sum

def is_perfect(number):
    return sum_of_divisors(number) == number
```

```python
# Example usage:
num = int(input("Enter a number: "))

if is_perfect(num):
    print(num, "is a perfect number.")
else:
    print(num, "is not a perfect number.")

def factorial(n):
    # Base case: factorial of 0 is 1
    if n == 0:
        return 1
    # Recursive case: factorial of n is n * factorial(n-1)
    else:
        return n * factorial(n - 1)

# Example usage:
num = int(input("Enter a number: "))

if num < 0:
    print("Factorial is not defined for negative numbers.")
else:
    print("Factorial of", num, "is:", factorial(num))

def fibonacci(n):
    # Initialize the first two terms of the Fibonacci series
    fib_series = [0, 1]

    # Generate subsequent terms of the Fibonacci series
    for i in range(2, n):
        next_term = fib_series[-1] + fib_series[-2]
        fib_series.append(next_term)

    return fib_series

# Example usage:
terms = int(input("Enter the number of terms in the Fibonacci series: "))

if terms <= 0:
    print("Please enter a positive integer.")
else:
    print("Fibonacci series up to", terms, "terms:")
    print(fibonacci(terms))

def is_prime(num):
    if num < 2:
        return False
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            return False
    return True

def primes_less_than_n(n):
    prime_list = []
    for i in range(2, n):
        if is_prime(i):
            prime_list.append(i)
    return prime_list

# Example usage:
limit = int(input("Enter a number: "))

print("Prime numbers less than", limit, "are:")
print(primes_less_than_n(limit))

def find_factors(number):
    factors = []
    for i in range(1, number):
        if number % i == 0:
            factors.append(i)
    return factors

def is_perfect(number):
    factors = find_factors(number)
    total = sum(factors)
    return total == number

# Example usage:
num = int(input("Enter a number: "))

factors = find_factors(num)
```

```
print("Factors of", num, "are:", factors)

if is_perfect(num):
    print(num, "is a perfect number.")
else:
    print(num, "is not a perfect number.")
```