PYTHON

100 QUIZZES

100 Quizzes from
Beginner to Advanced
with Detailed Solutions

2024

PYTHON MASTERY

Python Mastery: 100 Quizzes from Beginner to Advanced with Detailed Solutions

Introduction

Welcome to "100 Python Programming Language Quiz with Solutions"! Whether you're just starting your journey into the world of programming or looking to enhance your Python skills, this book is designed to be your companion from beginner to mastery level.

Python has emerged as one of the most popular and versatile programming languages in recent years. Its simplicity, readability, and vast ecosystem of libraries make it an ideal choice for beginners and seasoned developers alike. Whether you're interested in web development, data analysis, machine learning, or automation, Python has you covered.

This book is not just another Python tutorial; it's a hands-on learning experience designed to challenge and expand your understanding of Python concepts. Instead of passive reading, you'll actively engage with the

material through a series of quizzes covering a wide range of topics, from basic syntax to advanced concepts.

What Makes This Book Unique?

● **Comprehensive Coverage**: The quizzes in this book cover a diverse range of topics, ensuring that you get a well-rounded understanding of Python programming.

● **Progressive Difficulty**: Starting from simple concepts, the difficulty of the quizzes gradually increases, allowing you to build your skills incrementally.
● **Detailed Solutions**: Each quiz is accompanied by a detailed solution, providing step-by-step explanations to help you understand the reasoning behind the correct answer.
● **Hands-On Learning**: By actively solving problems, you'll reinforce your understanding of Python concepts and gain practical experience that will stick with you long after you've finished the book.
● **Suitable for All Levels**: Whether you're a complete beginner or an experienced developer looking to brush up on your skills, there's something in this book for everyone.

How to Use This Book

● **Start from the Beginning**: If you're new to Python programming, it's best to start from the beginning and work your way through the quizzes sequentially. This will ensure that you build a strong foundation before tackling more advanced topics.

● **Take Your Time**: Don't rush through the quizzes. Take the time to understand each question and solution thoroughly. Experiment with the code examples and try to come up with alternative solutions on your own.

● **Practice, Practice, Practice**: The key to mastering Python (or any programming language) is practice. Don't just read the book—make sure to write code, experiment, and solve problems on your own. The more you practice, the more confident you'll become in your abilities.

● **Use the Solutions Wisely**: While it's tempting to look at the solutions right away when you get stuck, try to solve the problem on your own first. Only refer to the solutions when you're truly stuck or after you've given it your best effort.

Whether you're looking to kickstart your career in software development, enhance your data analysis skills, or simply explore the fascinating world of programming, Python is an excellent choice. And "100 Python Programming Language Quiz with Solutions" is here to guide you every step of the way.

1. Question: Write a Python program to find the largest number among three numbers. **Prompt:** Enter three numbers separated by spaces:

Solution:

python

```
# Prompting user to enter three numbers separated by spaces num1, num2, num3 = map(int, input("Enter three numbers separated by spaces: ").split())

# Comparing numbers to find the largest if num1 >= num2 and num1 >= num3: print("The largest number is:", num1)

elif num2 >= num1 and num2 >= num3: print("The largest number is:", num2)
else:
print("The largest number is:", num3)
```

2. Question: Write a Python program to check if a year is a leap year or not. **Prompt:** Enter a year:

Solution:

python

```
# Prompting user to enter a year year = int(input("Enter a year: "))

# Checking if it's a leap year
if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0): print(year, "is a leap year.")
```

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

3. Question: Write a Python program to calculate the sum of natural numbers up to a given number.
Prompt: Enter a positive integer:

Solution:
python
```
# Prompting user to enter a positive integer n = int(input("Enter a positive integer: "))

# Calculating the sum of natural numbers sum = n * (n + 1) // 2
print("The sum of natural numbers up to", n, "is:", sum)
```
4. Question: Write a Python program to check if a number is a prime number or not. **Prompt:** Enter a number:

Solution:
python
```
# Prompting user to enter a number num = int(input("Enter a number: "))

# Checking if it's a prime number
if num > 1:
for i in range(2, int(num**0.5) + 1):

if (num % i) == 0:
print(num, "is not a prime number.") break

else:
print(num, "is a prime number.") else:
print(num, "is not a prime number.")
```
5. Question: Write a Python program to reverse a string. **Prompt:** Enter a string:

Solution:
python
```
# Prompting user to enter a string string = input("Enter a string: ")
```

```
# Reversing the string
reversed_string = string[::-1]
print("Reversed string:", reversed_string)
```

6. Question: Write a Python program to count the number of vowels in a string. **Prompt:** Enter a string:

Solution:

python

```
# Prompting user to enter a string string = input("Enter a string: ")

# Counting the number of vowels vowels = 'aeiouAEIOU'
count = 0
for char in string:

if char in vowels:
count += 1
print("Number of vowels:", count)
```

7. Question: Write a Python program to find the sum of all elements in a list. **Prompt:** Enter elements of the list separated by spaces:

Solution:

python

```
# Prompting user to enter elements of the list separated by spaces elements
= input("Enter elements of the list separated by spaces: ").split()

# Converting elements to integers and calculating the sum sum = 0
for element in elements:

sum += int(element)
print("Sum of elements:", sum)
```

8. Question: Write a Python program to find the factorial of a number using recursion. **Prompt:** Enter a non-negative integer:

Solution:

python

```
# Prompting user to enter a non-negative integer num = int(input("Enter a
non-negative integer: "))
```

```
# Defining factorial function using recursion def factorial(n):
if n == 0:

return 1
else:
return n * factorial(n - 1)

print("Factorial of", num, "is:", factorial(num))
```

9. Question: Write a Python program to find the Fibonacci series up to n terms. **Prompt:** Enter the number of terms:

Solution:

python

```
# Prompting user to enter the number of terms n = int(input("Enter the number of terms: "))

# Initializing first two terms first_term, second_term = 0, 1
fibonacci_series = [first_term, second_term]

# Generating Fibonacci series
for i in range(2, n):
next_term = first_term + second_term
fibonacci_series.append(next_term)
first_term, second_term = second_term, next_term

print("Fibonacci series up to", n, "terms:", fibonacci_series)
```

10. Question: Write a Python program to find the GCD (Greatest Common Divisor) of two numbers.
Prompt: Enter two positive integers separated by spaces:

Solution:

python

```
# Prompting user to enter two positive integers separated by spaces num1, num2 = map(int, input("Enter two positive integers separated by spaces: ").split())
```

```
# Defining function to find GCD def gcd(a, b):
while b:
a, b = b, a % b
return a
print("GCD of", num1, "and", num2, "is:", gcd(num1, num2))
```

11. Question: Write a Python program to check if a string is a palindrome or not. **Prompt:** Enter a string:
Solution: Python

```
# Prompting user to enter a string string = input("Enter a string: ") #
Checking if it's a palindrome if string == string[::-1]:

print("Palindrome")
else:
print("Not a palindrome")
```

12. Question: Write a Python program to find the square root of a number. **Prompt:** Enter a number:

Solution:
python

```
# Prompting user to enter a number num = float(input("Enter a number: "))

# Calculating square root sqrt = num ** 0.5
print("Square root of", num, "is:", sqrt)
```

13. Question: Write a Python program to sort a list of elements in ascending order. **Prompt:** Enter elements of the list separated by spaces:

Solution:
python

```
# Prompting user to enter elements of the list separated by spaces elements
= input("Enter elements of the list separated by spaces: ").split()

# Sorting the list elements.sort()
print("Sorted list in ascending order:", elements)
```

14. Question: Write a Python program to find the sum of digits of a number. **Prompt:** Enter a number:

Solution:
python
```
# Prompting user to enter a number
num = int(input("Enter a number: "))
# Calculating sum of digits
sum_of_digits = sum(int(digit) for digit in str(num))

print("Sum of digits of", num, "is:", sum_of_digits)
```
15. Question: Write a Python program to find the maximum depth of nested lists. **Prompt:** Enter elements of the nested list separated by spaces (e.g., 1 2 [3 4]):

Solution:
python
```
# Prompting user to enter elements of the nested list separated by spaces
nested_list_str = input("Enter elements of the nested list separated by
spaces: ")

# Evaluating the string to get the nested list nested_list =
eval(nested_list_str)
# Defining function to find maximum depth
def max_depth(lst):
if isinstance(lst, list):

return 1 + max(max_depth(item) for item in lst) else:
return 0

print("Maximum depth of the nested list:", max_depth(nested_list))
```

16. Question: Write a Python program to find the common elements between two lists. **Prompt:** Enter elements of the first list separated by spaces:
Prompt (contd.): Enter elements of the second list separated by spaces:

Solution:
python
```
# Prompting user to enter elements of the first list separated by spaces
list1 = input("Enter elements of the first list separated by spaces: ").split()
```

```
# Prompting user to enter elements of the second list separated by spaces
list2 = input("Enter elements of the second list separated by spaces:
").split()

# Finding common elements
common_elements = list(set(list1) & set(list2))
print("Common elements between the two lists:", common_elements)
```

17. Question: Write a Python program to count the number of words in a sentence. **Prompt:** Enter a sentence:

Solution:

python

```
# Prompting user to enter a sentence sentence = input("Enter a sentence: ")

# Counting the number of words word_count = len(sentence.split())
print("Number of words in the sentence:", word_count)
```

18. Question: Write a Python program to find the area of a triangle given its three sides. **Prompt:** Enter the lengths of the three sides of the triangle separated by spaces:

Solution:

python

```
import math

# Prompting user to enter the lengths of the three sides of the triangle
separated by spaces
a, b, c = map(float, input("Enter the lengths of the three sides of the triangle
separated by spaces: ").split())

# Calculating the semi-perimeter s = (a + b + c) / 2
# Calculating the area using Heron's formula area = math.sqrt(s * (s - a) * (s
- b) * (s - c))
print("Area of the triangle:", area)
```

19. Question: Write a Python program to generate a random number within a specified range.
Prompt: Enter the lower and upper bounds of the range separated by a space:

Solution:
python
```
import random

# Prompting user to enter the lower and upper bounds of the range separated by a space
lower, upper = map(int, input("Enter the lower and upper bounds of the range separated by a space: ").split())

# Generating a random number within the specified range random_num = random.randint(lower, upper)
print("Random number within the range:", random_num)
```

20. Question: Write a Python program to convert a decimal number to binary, octal, and hexadecimal formats.
Prompt: Enter a decimal number:

Solution:
python
```
# Prompting user to enter a decimal number decimal_num = int(input("Enter a decimal number: "))

# Converting to binary, octal, and hexadecimal formats binary_num = bin(decimal_num)
octal_num = oct(decimal_num)
hex_num = hex(decimal_num)

print("Binary:", binary_num) print("Octal:", octal_num)
print("Hexadecimal:", hex_num)
```

21. Question: Write a Python program to find the LCM (Least Common Multiple) of two numbers.
Prompt: Enter two positive integers separated by spaces:

Solution:
python
```
# Prompting user to enter two positive integers separated by spaces num1,
```

```
num2 = map(int, input("Enter two positive integers separated by spaces: ").split())

# Defining function to find LCM
def lcm(a, b):
gcd = lambda x, y: x if not y else gcd(y, x % y)
return abs(a * b) // gcd(a, b)
print("LCM of", num1, "and", num2, "is:", lcm(num1, num2))
```

22. Question: Write a Python program to find the area and circumference of a circle given its radius.
Prompt: Enter the radius of the circle:

Solution:
python
```
import math

# Prompting user to enter the radius of the circle radius = float(input("Enter the radius of the circle: "))

# Calculating area and circumference area = math.pi * radius ** 2
circumference = 2 * math.pi * radius

print("Area of the circle:", area)
print("Circumference of the circle:", circumference)
```

23. Question: Write a Python program to check if a number is an Armstrong number or not.
Prompt: Enter a number:

Solution:
python
```
# Prompting user to enter a number num = int(input("Enter a number: "))

# Calculating the number of digits num_str = str(num)
num_digits = len(num_str)
# Checking if it's an Armstrong number
sum = sum(int(digit) ** num_digits for digit in num_str) if sum == num:
```

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

24. Question: Write a Python program to find the median of three numbers.
Prompt: Enter three numbers separated by spaces:

Solution:
python
```
# Prompting user to enter three numbers separated by spaces num1, num2,
num3 = map(int, input("Enter three numbers separated by spaces: ").split())

# Finding median
median = sorted([num1, num2, num3])[1]
print("Median of the three numbers:", median)
```

25. Question: Write a Python program to find the sum of all even numbers up to a given number.
Prompt: Enter a positive integer:

Solution:
python
```
# Prompting user to enter a positive integer n = int(input("Enter a positive
integer: "))

# Calculating sum of even numbers sum = 0
for i in range(2, n + 1, 2):

sum += i
print("Sum of even numbers up to", n, "is:", sum)
```

26. Question: Write a Python program to count the number of occurrences of a specific character in a string.
Prompt: Enter a string:
Prompt (contd.): Enter the character to count:

Solution:
python

```
# Prompting user to enter a string string = input("Enter a string: ")

# Prompting user to enter the character to count char_to_count =
input("Enter the character to count: ")
# Counting occurrences of the character count =
string.count(char_to_count)
print("Number of occurrences of", char_to_count, "in the string:", count)
```

27. Question: Write a Python program to print the ASCII value of a character. **Prompt:** Enter a character:

Solution:

python

```
# Prompting user to enter a character char = input("Enter a character: ")

# Getting ASCII value ascii_value = ord(char)
print("ASCII value of", char, "is:", ascii_value)
```

28. Question: Write a Python program to find the sum of all prime numbers up to a given number.
Prompt: Enter a positive integer:

Solution:

python

```
# Prompting user to enter a positive integer n = int(input("Enter a positive
integer: "))

# Defining function to check if a number is prime def is_prime(num):
if num <= 1:

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

# Calculating sum of prime numbers
sum = sum(i for i in range(2, n + 1) if is_prime(i))
print("Sum of prime numbers up to", n, "is:", sum)
```

29. Question: Write a Python program to find the length of the last word in a sentence. **Prompt:** Enter a sentence:

Solution:
python
```
# Prompting user to enter a sentence sentence = input("Enter a sentence: ")

# Finding the length of the last word words = sentence.split()
if words:

last_word_length = len(words[-1]) else:
last_word_length = 0
print("Length of the last word in the sentence:", last_word_length)
```

30. Question: Write a Python program to find the smallest number among three numbers.
Prompt: Enter three numbers separated by spaces:

Solution:
python
```
# Prompting user to enter three numbers separated by spaces num1, num2,
num3 = map(int, input("Enter three numbers separated by spaces: ").split())

# Comparing numbers to find the smallest if num1 <= num2 and num1 <=
num3:
print("The smallest number is:", num1)

elif num2 <= num1 and num2 <= num3: print("The smallest number is:",
num2)
else:
print("The smallest number is:", num3)
```

31. Question: Write a Python program to find the sum of digits in a factorial of a number.
Prompt: Enter a non-negative integer:

Solution:
python

```
# Prompting user to enter a non-negative integer num = int(input("Enter a non-negative integer: "))

# Calculating factorial factorial = 1
for i in range(1, num + 1):

factorial *= i
# Calculating sum of digits in factorial sum_of_digits = sum(int(digit) for digit in str(factorial))
print("Sum of digits in factorial of", num, "is:", sum_of_digits)
```

32. Question: Write a Python program to check if a number is a perfect number or not. **Prompt:** Enter a positive integer:

Solution:

python

```
# Prompting user to enter a positive integer num = int(input("Enter a positive integer: "))

# Finding divisors and calculating sum divisors = [1]
for i in range(2, num // 2 + 1):

if num % i == 0:
divisors.append(i)
divisor_sum = sum(divisors)
# Checking if it's a perfect number if divisor_sum == num:
print(num, "is a perfect number.") else:
print(num, "is not a perfect number.")
```

33. Question: Write a Python program to convert a string to a list of words. **Prompt:** Enter a string:

Solution:

python

```
# Prompting user to enter a string string = input("Enter a string: ")

# Converting the string to a list of words words = string.split()
print("List of words:", words)
```

34. Question: Write a Python program to find the sum of all multiples of 3 or 5 below a given number.
Prompt: Enter a positive integer:

Solution:
python
```
# Prompting user to enter a positive integer n = int(input("Enter a positive integer: "))

# Calculating sum of multiples of 3 or 5
sum = sum(i for i in range(n) if i % 3 == 0 or i % 5 == 0)
print("Sum of multiples of 3 or 5 below", n, "is:", sum)
```
35. Question: Write a Python program to find the number of digits in a given number. **Prompt:** Enter a number:

Solution:
python
```
# Prompting user to enter a number num = int(input("Enter a number: "))

# Counting the number of digits num_digits = len(str(abs(num)))
print("Number of digits in", num, "is:", num_digits)
```

36. Question: Write a Python program to find the factorial of a number using an iterative approach.
Prompt: Enter a non-negative integer:

Solution:
python
```
# Prompting user to enter a non-negative integer num = int(input("Enter a non-negative integer: "))

# Calculating factorial using iteration factorial = 1
for i in range(1, num + 1):

factorial *= i
print("Factorial of", num, "is:", factorial)
```

37. Question: Write a Python program to find the number of days between two given dates.
Prompt: Enter the first date in YYYY-MM-DD format:
Prompt (contd.): Enter the second date in YYYY-MM-DD format:

Solution:
python
```
from datetime import datetime

# Prompting user to enter the first date
first_date_str = input("Enter the first date in YYYY-MM-DD format: ")
first_date = datetime.strptime(first_date_str, '%Y-%m-%d')

# Prompting user to enter the second date
second_date_str = input("Enter the second date in YYYY-MM-DD format: ")
second_date = datetime.strptime(second_date_str, '%Y-%m-%d')

# Calculating the difference in days
days_difference = abs((second_date - first_date).days) print("Number of days between the two dates:", days_difference)
```

38. Question: Write a Python program to convert temperature from Celsius to Fahrenheit and vice versa.
Prompt: Enter the temperature value:
Prompt (contd.): Enter 'C' for Celsius to Fahrenheit conversion or 'F' for Fahrenheit to Celsius conversion:

Solution:
python
```
# Prompting user to enter the temperature value temp_value = float(input("Enter the temperature value: "))

# Prompting user to choose conversion type
conversion_type = input("Enter 'C' for Celsius to Fahrenheit or 'F' for Fahrenheit to Celsius: ")
```

```
# Converting temperature
if conversion_type.upper() == 'C':
fahrenheit_temp = (temp_value * 9/5) + 32
print("Temperature in Fahrenheit:", fahrenheit_temp) elif
conversion_type.upper() == 'F':
celsius_temp = (temp_value - 32) * 5/9
print("Temperature in Celsius:", celsius_temp) else:
print("Invalid input!")
```

39. Question: Write a Python program to find the HCF (Highest Common Factor) of two numbers.
Prompt: Enter two positive integers separated by spaces:

Solution:
python

```
# Prompting user to enter two positive integers separated by spaces num1,
num2 = map(int, input("Enter two positive integers separated by spaces:
").split())
# Defining function to find HCF def hcf(a, b):

while b:
a, b = b, a % b
return a

print("HCF of", num1, "and", num2, "is:", hcf(num1, num2))
```

40. Question: Write a Python program to find the sum of all prime numbers within a given range.
Prompt: Enter the lower and upper bounds of the range separated by spaces:

Solution:
python

```
# Prompting user to enter the lower and upper bounds of the range
separated by spaces
lower, upper = map(int, input("Enter the lower and upper bounds of the
range separated by spaces: ").split())
```

```
# Defining function to check if a number is prime def is_prime(num):
if num <= 1:

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

# Calculating sum of prime numbers within the range sum = sum(i for i in
range(lower, upper + 1) if is_prime(i))
print("Sum of prime numbers within the range:", sum)
```

41. Question: Write a Python program to generate a list of all prime numbers within a given range.
Prompt: Enter the lower and upper bounds of the range separated by spaces:

Solution:
python

```
# Prompting user to enter the lower and upper bounds of the range
separated by spaces
lower, upper = map(int, input("Enter the lower and upper bounds of the
range separated by spaces: ").split())

# Defining function to check if a number is prime def is_prime(num):
if num <= 1:

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

# Generating list of prime numbers within the range
prime_numbers = [i for i in range(lower, upper + 1) if is_prime(i)]
print("Prime numbers within the range:", prime_numbers)
```

42. Question: Write a Python program to check if a string is an anagram of another string.
Prompt: Enter the first string:
Prompt (contd.): Enter the second string:

Solution:
python
Prompting user to enter the first string string1 = input("Enter the first string: ")

Prompting user to enter the second string string2 = input("Enter the second string: ") # Checking if they are anagrams if sorted(string1) == sorted(string2):

print("Anagrams")
else:
print("Not anagrams")

43. Question: Write a Python program to find all factors of a given number.
Prompt: Enter a positive integer:

Solution:
python
Prompting user to enter a positive integer num = int(input("Enter a positive integer: "))

# Finding factors
factors = [i for i in range(1, num + 1) if num % i == 0]
print("Factors of", num, "are:", factors)

44. Question: Write a Python program to check if a number is a palindrome or not using recursion.
Prompt: Enter a number:

Solution:
python
Prompting user to enter a number num = input("Enter a number: ")

```
# Defining function to check if it's a palindrome def is_palindrome(s):
if len(s) <= 1:
return True
elif s[0] == s[-1]:
return is_palindrome(s[1:-1]) else:
return False

if is_palindrome(num):
print("Palindrome")
else:
print("Not a palindrome")
```

45. Question: Write a Python program to check if a number is a perfect square or not. **Prompt:** Enter a positive integer:

Solution:

python

```
# Prompting user to enter a positive integer num = int(input("Enter a
positive integer: "))

# Checking if it's a perfect square if (num ** 0.5).is_integer():
print(num, "is a perfect square.") else:
print(num, "is not a perfect square.")
```

46. Question: Write a Python program to find the area of a rectangle given its length and width.
Prompt: Enter the length and width of the rectangle separated by spaces:

Solution:

python

```
# Prompting user to enter the length and width of the rectangle separated by
spaces
length, width = map(float, input("Enter the length and width of the
rectangle separated by spaces: ").split())

# Calculating area area = length * width
print("Area of the rectangle:", area)
```

47. Question: Write a Python program to find the area of a triangle given its base and height.
Prompt: Enter the base and height of the triangle separated by spaces:

Solution:
python
```
# Prompting user to enter the base and height of the triangle separated by spaces
base, height = map(float, input("Enter the base and height of the triangle separated by spaces: ").split())

# Calculating area
area = 0.5 * base * height
print("Area of the triangle:", area)
```

48. Question: Write a Python program to find the area of a trapezoid given its two bases and height.
Prompt: Enter the lengths of the two bases and the height of the trapezoid separated by spaces:

Solution:
python
```
# Prompting user to enter the lengths of the two bases and the height of the trapezoid separated by spaces
base1, base2, height = map(float, input("Enter the lengths of the two bases and the height of the trapezoid separated by spaces: ").split())

# Calculating area
area = 0.5 * (base1 + base2) * height
print("Area of the trapezoid:", area)
```

49. Question: Write a Python program to find the surface area and volume of a cylinder given its radius and height.
Prompt: Enter the radius and height of the cylinder separated by spaces:

Solution:
python
```
import math
```

```
# Prompting user to enter the radius and height of the cylinder separated by spaces
radius, height = map(float, input("Enter the radius and height of the cylinder separated by spaces: ").split())

# Calculating surface area and volume
surface_area = 2 * math.pi * radius * (radius + height) volume = math.pi * radius ** 2 * height

print("Surface area of the cylinder:", surface_area) print("Volume of the cylinder:", volume)
```

50. Question: Write a Python program to find the area of a sphere given its radius. **Prompt:** Enter the radius of the sphere:

Solution:

python

```
import math

# Prompting user to enter the radius of the sphere radius = float(input("Enter the radius of the sphere: "))
# Calculating area
area = 4 * math.pi * radius ** 2
print("Area of the sphere:", area)
```

51. Question: Write a Python program to find the volume of a sphere given its radius. **Prompt:** Enter the radius of the sphere:

Solution:

python

```
import math

# Prompting user to enter the radius of the sphere radius = float(input("Enter the radius of the sphere: "))
# Calculating volume
volume = (4/3) * math.pi * radius ** 3
print("Volume of the sphere:", volume)
```

52. Question: Write a Python program to find the sum of the first n natural numbers using recursion.

Prompt: Enter a positive integer:

Solution:
python
```
# Prompting user to enter a positive integer n = int(input("Enter a positive integer: "))

# Defining function to find the sum of first n natural numbers def sum_of_natural_numbers(n):
if n == 1:

return 1
else:
return n + sum_of_natural_numbers(n - 1)

print("Sum of first", n, "natural numbers:", sum_of_natural_numbers(n))
```

53. Question: Write a Python program to find the sum of digits of a number using recursion.
Prompt: Enter a positive integer:

Solution:
python
```
# Prompting user to enter a positive integer num = int(input("Enter a positive integer: "))

# Defining function to find the sum of digits def sum_of_digits(n):
if n == 0:

return 0
else:
return n % 10 + sum_of_digits(n // 10)

print("Sum of digits of", num, "is:", sum_of_digits(num))
```
54. Question: Write a Python program to check if a number is a strong number or not. **Prompt:** Enter a positive integer:

Solution:
python
```
# Prompting user to enter a positive integer num = int(input("Enter a positive integer: "))

# Defining function to find factorial of a number def factorial(n):
if n == 0:

return 1
else:
return n * factorial(n - 1)

# Checking if it's a strong number
sum_of_factorial_digits = sum(factorial(int(digit)) for digit in str(num))
if sum_of_factorial_digits == num:
print(num, "is a strong number.")

else:
print(num, "is not a strong number.")
```
55. Question: Write a Python program to find the length of a string using recursion. **Prompt:** Enter a string:

Solution:
python
```
# Prompting user to enter a string string = input("Enter a string: ")

# Defining function to find the length of the string def string_length(s):
if s == '':

return 0
else:
return 1 + string_length(s[1:])

print("Length of the string:", string_length(string))
```
56. Question: Write a Python program to count the occurrences of a substring in a given string.

Prompt: Enter a string:
Prompt (contd.): Enter the substring to count:

Solution:
python
```
# Prompting user to enter a string string = input("Enter a string: ")

# Prompting user to enter the substring to count substring = input("Enter the substring to count: ")
# Counting occurrences of the substring count = string.count(substring)
print("Number of occurrences of", substring, "in the string:", count)
```
57. Question: Write a Python program to count the number of vowels in a given string. **Prompt:** Enter a string:

Solution:
python
```
# Prompting user to enter a string string = input("Enter a string: ")

# Counting the number of vowels
vowels = 'aeiouAEIOU'
vowel_count = sum(1 for char in string if char in vowels)

print("Number of vowels in the string:", vowel_count)
```

58. Question: Write a Python program to find the frequency of each character in a given string.
Prompt: Enter a string:

Solution:
python
```
# Prompting user to enter a string string = input("Enter a string: ")

# Finding frequency of each character
char_freq = {}
for char in string:

char_freq[char] = char_freq.get(char, 0) + 1
print("Frequency of each character:") for char, freq in char_freq.items():
```

```
print(char, ":", freq)
```

59. Question: Write a Python program to find the largest and smallest word in a given string.
Prompt: Enter a string:

Solution:
python
```
# Prompting user to enter a string string = input("Enter a string: ")

# Finding the largest and smallest word words = string.split()
largest_word = max(words, key=len) smallest_word = min(words, key=len)

print("Largest word in the string:", largest_word) print("Smallest word in the string:", smallest_word)
```

60. Question: Write a Python program to find the number of uppercase, lowercase, and numeric characters in a given string.
Prompt: Enter a string:

Solution:
python
```
# Prompting user to enter a string string = input("Enter a string: ")

# Counting the number of uppercase, lowercase, and numeric characters
uppercase_count = sum(1 for char in string if char.isupper())
lowercase_count = sum(1 for char in string if char.islower()) numeric_count = sum(1 for char in string if char.isdigit())

print("Number of uppercase characters:", uppercase_count) print("Number of lowercase characters:", lowercase_count) print("Number of numeric characters:", numeric_count)
```

61. Question: Write a Python program to find the length of the longest consecutive sequence of a character in a given string.
Prompt: Enter a string:

Solution:
python
```
# Prompting user to enter a string string = input("Enter a string: ")

# Finding the length of the longest consecutive sequence of a character
max_length = 1
current_length = 1
for i in range(1, len(string)):

if string[i] == string[i - 1]:
current_length += 1
max_length = max(max_length, current_length)

else:
current_length = 1
print("Length of the longest consecutive sequence of a character:",
max_length)
```
62. Question: Write a Python program to remove all duplicates from a given list. **Prompt:** Enter elements of the list separated by spaces:

Solution:
python
```
# Prompting user to enter elements of the list separated by spaces elements
= input("Enter elements of the list separated by spaces: ").split()

# Removing duplicates
unique_elements = list(set(elements))
print("List after removing duplicates:", unique_elements)
```
63. Question: Write a Python program to find the second largest number in a given list. **Prompt:** Enter elements of the list separated by spaces:

Solution:
python
```
# Prompting user to enter elements of the list separated by spaces elements
= list(map(int, input("Enter elements of the list separated by spaces:
").split()))
```

```
# Finding the second largest number
unique_elements = list(set(elements))
if len(unique_elements) < 2:

print("Second largest number does not exist.") else:
unique_elements.sort()
second_largest = unique_elements[-2]
print("Second largest number:", second_largest)
```

64. Question: Write a Python program to find the maximum product of two integers in a given list of integers.
Prompt: Enter elements of the list separated by spaces:

Solution:
python

```
# Prompting user to enter elements of the list separated by spaces elements
= list(map(int, input("Enter elements of the list separated by spaces:
").split()))

# Finding the maximum product of two integers elements.sort()
max_product = elements[-1] * elements[-2]

print("Maximum product of two integers in the list:", max_product)
```

65. Question: Write a Python program to check if a given list is sorted in non-decreasing order.
Prompt: Enter elements of the list separated by spaces:

Solution:
python

```
# Prompting user to enter elements of the list separated by spaces elements
= list(map(int, input("Enter elements of the list separated by spaces:
").split()))

# Checking if the list is sorted in non-decreasing order sorted_order =
all(elements[i] <= elements[i + 1] for i in range(len(elements) - 1))
```

```
if sorted_order:
print("List is sorted in non-decreasing order.")
else:
print("List is not sorted in non-decreasing order.")
```

66. Question: Write a Python program to merge two sorted lists into a single sorted list. **Prompt:** Enter elements of the first sorted list separated by spaces:
Prompt (contd.): Enter elements of the second sorted list separated by spaces:

Solution:
python
```
# Prompting user to enter elements of the first sorted list separated by spaces
list1 = list(map(int, input("Enter elements of the first sorted list separated by spaces: ").split()))

# Prompting user to enter elements of the second sorted list separated by spaces
list2 = list(map(int, input("Enter elements of the second sorted list separated by spaces: ").split()))

# Merging the two lists
merged_list = sorted(list1 + list2) print("Merged sorted list:", merged_list)
```
67. Question: Write a Python program to reverse a given list. **Prompt:** Enter elements of the list separated by spaces:

Solution:
python
```
# Prompting user to enter elements of the list separated by spaces elements = list(map(int, input("Enter elements of the list separated by spaces: ").split()))

# Reversing the list
reversed_list = elements[::-1]
print("Reversed list:", reversed_list)
```

68. Question: Write a Python program to remove all occurrences of a given element from a list.
Prompt: Enter elements of the list separated by spaces:
Prompt (contd.): Enter the element to remove:

Solution:
python
```
# Prompting user to enter elements of the list separated by spaces elements = list(map(int, input("Enter elements of the list separated by spaces: ").split()))

# Prompting user to enter the element to remove
element_to_remove = int(input("Enter the element to remove: "))

# Removing all occurrences of the given element
filtered_list = [element for element in elements if element != element_to_remove]
print("List after removing all occurrences of", element_to_remove, ":", filtered_list)
```

69. Question: Write a Python program to rotate a given list left by n elements. **Prompt:** Enter elements of the list separated by spaces:
Prompt (contd.): Enter the value of n:

Solution:
python
```
# Prompting user to enter elements of the list separated by spaces elements = list(map(int, input("Enter elements of the list separated by spaces: ").split()))

# Prompting user to enter the value of n n = int(input("Enter the value of n: "))
# Rotating the list left by n elements rotated_list = elements[n:] + elements[:n]
print("List after rotating left by", n, "elements:", rotated_list)
```

70. Question: Write a Python program to sort a list of tuples by the second element of each tuple.

Prompt: Enter elements of the list of tuples (each tuple separated by spaces and elements within tuple separated by commas):

Solution:
python
```
# Prompting user to enter elements of the list of tuples tuples =
[tuple(map(int, input("Enter elements of a tuple separated by commas:
").split(','))) for _ in range(int(input("Enter number of tuples: ")))]

# Sorting the list of tuples by the second element of each tuple sorted_tuples
= sorted(tuples, key=lambda x: x[1])
print("List of tuples sorted by the second element of each tuple:",
sorted_tuples)
```

71. Question: Write a Python program to find the sum of all elements in a matrix. **Prompt:** Enter the number of rows and columns of the matrix separated by spaces: **Prompt (contd.):** Enter elements of the matrix row-wise (each row separated by spaces):

Solution:
python
```
# Prompting user to enter the number of rows and columns of the matrix
separated by spaces
rows, cols = map(int, input("Enter the number of rows and columns of the
matrix separated by spaces: ").split())

# Prompting user to enter elements of the matrix row-wise matrix =
[list(map(int, input("Enter elements of row {}: ".format(i + 1)).split())) for i
in range(rows)]

# Calculating the sum of all elements in the matrix matrix_sum =
sum(sum(row) for row in matrix)
print("Sum of all elements in the matrix:", matrix_sum)
```

72. Question: Write a Python program to multiply two matrices.
Prompt: Enter the number of rows and columns of the first matrix separated by spaces: **Prompt (contd.):** Enter elements of the first matrix row-wise (each row separated by spaces):

Prompt (contd.): Enter the number of rows and columns of the second matrix separated by spaces:
Prompt (contd.): Enter elements of the second matrix row-wise (each row separated by spaces):

```
# Prompting user to enter the number of rows and columns of the first matrix separated by spaces
rows1, cols1 = map(int, input("Enter the number of rows and columns of the first matrix separated by spaces: ").split())

# Prompting user to enter elements of the first matrix row-wise matrix1 = [list(map(int, input("Enter elements of row {}: ".format(i + 1)).split())) for i in range(rows1)]

# Prompting user to enter the number of rows and columns of the second matrix separated by spaces
rows2, cols2 = map(int, input("Enter the number of rows and columns of the second matrix separated by spaces: ").split())

# Prompting user to enter elements of the second matrix row-wise matrix2 = [list(map(int, input("Enter elements of row {}: ".format(i + 1)).split())) for i in range(rows2)]

# Checking if matrices can be multiplied
if cols1 != rows2:
print("Matrices cannot be multiplied.")

else:
# Multiplying the matrices
result = [[sum(matrix1[i][k] * matrix2[k][j] for k in

range(cols1)) for j in range(cols2)] for i in range(rows1)]

print("Result of matrix multiplication:") for row in result:
print(row)
```

73. Question: Write a Python program to transpose a matrix.
Prompt: Enter the number of rows and columns of the matrix separated by

spaces: **Prompt (contd.):** Enter elements of the matrix row-wise (each row separated by spaces):

```
# Prompting user to enter the number of rows and columns of the matrix separated by spaces
rows, cols = map(int, input("Enter the number of rows and columns of the matrix separated by spaces: ").split())

# Prompting user to enter elements of the matrix row-wise matrix = [list(map(int, input("Enter elements of row {}: ".format(i + 1)).split())) for i in range(rows)]

# Transposing the matrix
transposed_matrix = [[matrix[j][i] for j in range(rows)] for i in range(cols)]

print("Transposed matrix:") for row in transposed_matrix: print(row)
```

74. Question: Write a Python program to find the sum of all elements in each column of a matrix.
Prompt: Enter the number of rows and columns of the matrix separated by spaces: **Prompt (contd.):** Enter elements of the matrix row-wise (each row separated by spaces):

Solution:
python

```
# Prompting user to enter the number of rows and columns of the matrix separated by spaces
rows, cols = map(int, input("Enter the number of rows and columns of the matrix separated by spaces: ").split())

# Prompting user to enter elements of the matrix row-wise matrix = [list(map(int, input("Enter elements of row {}: ".format(i + 1)).split())) for i in range(rows)]

# Finding the sum of all elements in each column of the matrix
column_sums = [sum(matrix[j][i] for j in range(rows)) for i in range(cols)]
print("Sum of all elements in each column:") for i, column_sum in enumerate(column_sums):
```

```
print("Column", i + 1, ":", column_sum)
```

75. Question: Write a Python program to find the sum of all elements in each row of a matrix.
Prompt: Enter the number of rows and columns of the matrix separated by spaces: **Prompt (contd.):** Enter elements of the matrix row-wise (each row separated by spaces):

Solution:
python
```
# Prompting user to enter the number of rows and columns of the matrix separated by spaces
rows, cols = map(int, input("Enter the number of rows and columns of the matrix separated by spaces: ").split())

# Prompting user to enter elements of the matrix row-wise matrix = [list(map(int, input("Enter elements of row {}: ".format(i + 1)).split())) for i in range(rows)]

# Finding the sum of all elements in each row of the matrix row_sums = [sum(row) for row in matrix]
print("Sum of all elements in each row:") for i, row_sum in enumerate(row_sums): print("Row", i + 1, ":", row_sum)
```

76. Question: Write a Python program to find the maximum element in each row of a matrix.
Prompt: Enter the number of rows and columns of the matrix separated by spaces: **Prompt (contd.):** Enter elements of the matrix row-wise (each row separated by spaces):

Solution:
python
```
# Prompting user to enter the number of rows and columns of the matrix separated by spaces
rows, cols = map(int, input("Enter the number of rows and columns of the matrix separated by spaces: ").split())
```

```
# Prompting user to enter elements of the matrix row-wise matrix = [list(map(int, input("Enter elements of row {}: ".format(i + 1)).split())) for i in range(rows)]

# Finding the maximum element in each row of the matrix row_max = [max(row) for row in matrix]
print("Maximum element in each row:") for i, max_element in enumerate(row_max): print("Row", i + 1, ":", max_element)
```

77. Question: Write a Python program to find the maximum element in each column of a matrix.
Prompt: Enter the number of rows and columns of the matrix separated by spaces: **Prompt (contd.):** Enter elements of the matrix row-wise (each row separated by spaces):

Solution:
python

```
# Prompting user to enter the number of rows and columns of the matrix separated by spaces
rows, cols = map(int, input("Enter the number of rows and columns of the matrix separated by spaces: ").split())

# Prompting user to enter elements of the matrix row-wise matrix = [list(map(int, input("Enter elements of row {}: ".format(i + 1)).split())) for i in range(rows)]

# Finding the maximum element in each column of the matrix col_max = [max(matrix[j][i] for j in range(rows)) for i in range(cols)]

print("Maximum element in each column:") for i, max_element in enumerate(col_max): print("Column", i + 1, ":", max_element)
```

78. Question: Write a Python program to find the minimum element in each row of a matrix.
Prompt: Enter the number of rows and columns of the matrix separated by spaces: **Prompt (contd.):** Enter elements of the matrix row-wise (each row separated by spaces):

Solution:
python
```
# Prompting user to enter the number of rows and columns of the matrix separated by spaces
rows, cols = map(int, input("Enter the number of rows and columns of the matrix separated by spaces: ").split())

# Prompting user to enter elements of the matrix row-wise matrix = [list(map(int, input("Enter elements of row {}: ".format(i + 1)).split())) for i in range(rows)]

# Finding the minimum element in each row of the matrix row_min = [min(row) for row in matrix]
print("Minimum element in each row:") for i, min_element in enumerate(row_min): print("Row", i + 1, ":", min_element)
```

79. Question: Write a Python program to find the minimum element in each column of a matrix.
Prompt: Enter the number of rows and columns of the matrix separated by spaces: **Prompt (contd.):** Enter elements of the matrix row-wise (each row separated by spaces):

Solution:
python
```
# Prompting user to enter the number of rows and columns of the matrix separated by spaces
rows, cols = map(int, input("Enter the number of rows and columns of the matrix separated by spaces: ").split())

# Prompting user to enter elements of the matrix row-wise matrix = [list(map(int, input("Enter elements of row {}: ".format(i + 1)).split())) for i in range(rows)]

# Finding the minimum element in each column of the matrix col_min = [min(matrix[j][i] for j in range(rows)) for i in range(cols)]

print("Minimum element in each column:") for i, min_element in enumerate(col_min): print("Column", i + 1, ":", min_element)
```

80. Question: Write a Python program to find the frequency of each element in a given list.
Prompt: Enter elements of the list separated by spaces:

Solution:
python
```
# Prompting user to enter elements of the list separated by spaces elements = input("Enter elements of the list separated by spaces: ").split()

# Finding the frequency of each element element_freq = {}
for element in elements:
element_freq[element] = element_freq.get(element, 0) + 1
print("Frequency of each element:") for element, freq in element_freq.items(): print(element, ":", freq)
```

81. Question: Write a Python program to find the frequency of each element in a given list using collections module.
Prompt: Enter elements of the list separated by spaces:

Solution:
python
```
from collections import Counter

# Prompting user to enter elements of the list separated by spaces elements = input("Enter elements of the list separated by spaces: ").split()

# Finding the frequency of each element using Counter element_freq = Counter(elements)
print("Frequency of each element:") for element, freq in element_freq.items(): print(element, ":", freq)
```

82. Question: Write a Python program to find the first non-repeating character in a given string.
Prompt: Enter a string:

Solution:
python
```
from collections import Counter
```

```
# Prompting user to enter a string string = input("Enter a string: ") # Finding the first non-repeating character char_freq = Counter(string)
for char in string:

if char_freq[char] == 1:
print("First non-repeating character:", char) break

else:
print("No non-repeating character found.")
```

83. Question: Write a Python program to find the first non-repeating character in a given string using OrderedDict.
Prompt: Enter a string:

Solution:
python

```
from collections import OrderedDict

# Prompting user to enter a string string = input("Enter a string: ")

# Finding the first non-repeating character char_freq = OrderedDict()
for char in string:

char_freq[char] = char_freq.get(char, 0) + 1

for char, freq in char_freq.items():
if freq == 1:
print("First non-repeating character:", char) break

else:
print("No non-repeating character found.")
```

84. Question: Write a Python program to find the first non-repeating character in a given string using a dictionary.
Prompt: Enter a string:

Solution:
python

```
# Prompting user to enter a string string = input("Enter a string: ")
```

```
# Finding the first non-repeating character char_freq = {}
for char in string:

char_freq[char] = char_freq.get(char, 0) + 1

for char in string:
if char_freq[char] == 1:
print("First non-repeating character:", char) break

else:
print("No non-repeating character found.")
```

85. Question: Write a Python program to find all duplicate characters in a given string. **Prompt:** Enter a string:

Solution:

python

```
# Prompting user to enter a string string = input("Enter a string: ")

# Finding all duplicate characters
char_freq = {}
for char in string:

char_freq[char] = char_freq.get(char, 0) + 1
duplicates = [char for char, freq in char_freq.items() if freq > 1]
print("Duplicate characters:", duplicates)
```

86. Question: Write a Python program to find all duplicate characters in a given string using Counter.
Prompt: Enter a string:

Solution:

python

```
from collections import Counter

# Prompting user to enter a string string = input("Enter a string: ")

# Finding all duplicate characters using Counter
char_freq = Counter(string)
```

```
duplicates = [char for char, freq in char_freq.items() if freq > 1]

print("Duplicate characters:", duplicates)
```

87. Question: Write a Python program to find all duplicate characters in a given string using OrderedDict.
Prompt: Enter a string:

Solution:
python
```
from collections import OrderedDict

# Prompting user to enter a string string = input("Enter a string: ")

# Finding all duplicate characters using OrderedDict char_freq =
OrderedDict()
for char in string:

char_freq[char] = char_freq.get(char, 0) + 1
duplicates = [char for char, freq in char_freq.items() if freq > 1]
print("Duplicate characters:", duplicates)
```

88. Question: Write a Python program to remove all duplicate characters from a given string.
Prompt: Enter a string:

Solution:
python
```
# Prompting user to enter a string string = input("Enter a string: ")

# Removing all duplicate characters
unique_chars = ''.join(OrderedDict.fromkeys(string))
print("String after removing duplicates:", unique_chars)
```

89. Question: Write a Python program to remove all duplicate characters from a given string using Counter.
Prompt: Enter a string:

Solution:
python
```
from collections import Counter

# Prompting user to enter a string string = input("Enter a string: ")

# Removing all duplicate characters using Counter
unique_chars = ''.join(char for char, freq in Counter(string).items() if freq == 1)

print("String after removing duplicates:", unique_chars)
```

90. Question: Write a Python program to remove all duplicate characters from a given string using OrderedDict.
Prompt: Enter a string:

Solution:
python
```
from collections import OrderedDict

# Prompting user to enter a string string = input("Enter a string: ")
# Removing all duplicate characters using OrderedDict unique_chars = ''.join(OrderedDict.fromkeys(string))
print("String after removing duplicates:", unique_chars)
```
91. Question: Write a Python program to find the median of three given numbers. **Prompt:** Enter three numbers separated by spaces:
Solution: python

```
# Prompting user to enter three numbers separated by spaces nums = list(map(int, input("Enter three numbers separated by spaces: ").split()))

# Sorting the numbers nums.sort()
# Finding the median median = nums[1]
print("Median of the three numbers:", median)
```

92. Question: Write a Python program to check if two strings are anagrams of each other. **Prompt:** Enter the first string:
Prompt (contd.): Enter the second string:

Solution:
python
```
# Prompting user to enter the first string str1 = input("Enter the first string:
")

# Prompting user to enter the second string str2 = input("Enter the second
string: ")
# Checking if the strings are anagrams if sorted(str1) == sorted(str2):
print("The strings are anagrams.") else:
print("The strings are not anagrams.")
```
93. Question: Write a Python program to find the next prime number after a given number. **Prompt:** Enter a number:

Solution:
python
```
# Prompting user to enter a number num = int(input("Enter a number: "))

# Finding the next prime number after the given number def is_prime(n):
if n <= 1:

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

next_prime = num + 1 while True:
if is_prime(next_prime):
print("Next prime number after", num, ":", next_prime) break
next_prime += 1
```
94. Question: Write a Python program to find the number of trailing zeros in the factorial of a given number. **Prompt:** Enter a number:

Solution:
python
```
# Prompting user to enter a number num = int(input("Enter a number: "))
```

```
# Finding the number of trailing zeros in the factorial def
trailing_zeros_factorial(n):
count = 0
i = 5
while n // i >= 1:
count += n // i
i *= 5
return count

print("Number of trailing zeros in the factorial of", num, ":",
trailing_zeros_factorial(num))
```

95. Question: Write a Python program to check if a given number is a palindrome. **Prompt:** Enter a number:

Solution:

python

```
# Prompting user to enter a number num = input("Enter a number: ")

# Checking if the number is a palindrome if num == num[::-1]:
print(num, "is a palindrome.") else:
print(num, "is not a palindrome.")
```

96. Question: Write a Python program to generate all prime numbers up to a given number. **Prompt:** Enter a number:

Solution:

python

```
# Prompting user to enter a number num = int(input("Enter a number: "))

# Generating all prime numbers up to the given number def
generate_primes(n):
primes = []
for i in range(2, n + 1):
for j in range(2, int(i ** 0.5) + 1): if i % j == 0:
break
else:
primes.append(i)
return primes
```

```
print("Prime numbers up to", num, ":", generate_primes(num))
```

97. Question: Write a Python program to find the GCD (Greatest Common Divisor) of two numbers. **Prompt:** Enter the first number:
Prompt (contd.): Enter the second number:

Solution:
python
```
# Prompting user to enter the first number num1 = int(input("Enter the first number: "))

# Prompting user to enter the second number num2 = int(input("Enter the second number: ")) # Finding the GCD of the two numbers def gcd(a, b):

while b != 0:
a, b = b, a % b
return a

print("GCD of", num1, "and", num2, ":", gcd(num1, num2))
```

98. Question: Write a Python program to find the LCM (Least Common Multiple) of two numbers. **Prompt:** Enter the first number:
Prompt (contd.): Enter the second number:

Solution:
python
```
# Prompting user to enter the first number num1 = int(input("Enter the first number: "))

# Prompting user to enter the second number num2 = int(input("Enter the second number: "))
# Finding the LCM of the two numbers def lcm(a, b):
return (a * b) // gcd(a, b) print("LCM of", num1, "and", num2, ":", lcm(num1, num2))
```
99. Question: Write a Python program to find all the divisors of a given number. **Prompt:** Enter a number:

Solution:
python
Prompting user to enter a number num = int(input("Enter a number: "))

# Finding all the divisors of the given number def find_divisors(n):
divisors = []
for i in range(1, n + 1):

if n % i == 0:
divisors.append(i) return divisors
print("Divisors of", num, ":", find_divisors(num))
100. **Question:** Write a Python program to find the sum of all divisors of a given number. **Prompt:** Enter a number:

Solution: ```python # Prompting user to enter a number num = int(input("Enter a number: ")) bash
Finding the sum of all divisors of the given number

arduino
def sum_of_divisors(n):
div_sum = 0
for i in range(1, n + 1): if n % i == 0:
div_sum += i
return div_sum

print("Sum of all divisors of", num, ":", sum_of_divisors(num)) ```

The End