Interview Questions  
python

11. Implement a program to find the Fibonacci series up to a given number.

Solution- def fibonacci\_series\_up\_to(n):

# Check if the input is non-positive

if n <= 0:

return []

# Initialize the first two numbers in the Fibonacci series

fib\_series = [0, 1]

# Generate the Fibonacci series up to the given number

while True:

next\_number = fib\_series[-1] + fib\_series[-2]

if next\_number > n:

break

fib\_series.append(next\_number)

return fib\_series

# Example usage

number = int(input("Enter a number: "))

fib\_series = fibonacci\_series\_up\_to(number)

print("Fibonacci series up to", number, ":", fib\_series)

12. Create a Python function to find the GCD (Greatest Common Divisor) of two numbers.

Solution- def gcd(a, b):

while b != 0:

a, b = b, a % b

return a

# Example usage

num1 = int(input("Enter the first number: "))

num2 = int(input("Enter the second number: "))

result = gcd(num1, num2)

print("The GCD of", num1, "and", num2, "is", result)

13. Write a program to count the number of vowels in a given string.

Solution def count\_vowels(input\_string):

# Define a set of vowels

vowels = set("aeiouAEIOU")

# Initialize a counter for vowels

count = 0

# Loop through each character in the string

for char in input\_string:

if char in vowels:

count += 1

return count

# Example usage

input\_string = input("Enter a string: ")

vowel\_count = count\_vowels(input\_string)

print("Number of vowels in the given string:", vowel\_count)

14. Implement a function to find the square root of a number without using any built-in library.

Solution- def sqrt(number, tolerance=1e-10):

if number < 0:

raise ValueError("Cannot compute the square root of a negative number")

if number == 0 or number == 1:

return number

# Initial guess will be the number itself

guess = number

# Keep improving the guess until the change is within the tolerance

while True:

# Apply Newton-Raphson formula

next\_guess = 0.5 \* (guess + number / guess)

# Check for convergence within the tolerance level

if abs(next\_guess - guess) < tolerance:

return next\_guess

guess = next\_guess

# Example usage

num = float(input("Enter a number: "))

try:

result = sqrt(num)

print(f"The square root of {num} is approximately {result}")

except ValueError as e:

print(e)

15. Create a Python program to sort a list of strings based on their lengths.

Solution- def sort\_by\_length(strings):

# Sort the list of strings by their lengths

return sorted(strings, key=len)

# Example usage

strings = ["apple", "banana", "kiwi", "grapefruit", "blueberry", "fig"]

sorted\_strings = sort\_by\_length(strings)

print("Sorted strings by length:", sorted\_strings)