**Question** : Write a Python program to print "Hello, World!".

# Explanation

print("Hello, World!")

**Question** : Calculate the sum of two numbers entered by the user.

# Explanation

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

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

sum\_result = num1 + num2

print("Sum:", sum\_result)

**Question** : Write a Python program to check if a given number is even or odd.

# Explanation

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

if num % 2 == 0:

  print("Even")

else:

  print("Odd")

**Question** : Calculate the factorial of a given number.

# Explanation

def factorial(n):

  if n == 0 or n == 1:

    return 1

  else:

    return n \* factorial(n - 1)

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

print("Factorial:", factorial(num))

**Question**: Find the largest among three numbers entered by the user.

# Explanation

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

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

num3 = float(input("Enter the third number: "))

largest = max(num1, num2, num3)

print("Largest number:", largest)

**Question** : Write a Python program to print all even numbers from 1 to 20.

# Explanation

for i in range(2, 21, 2):

  print(i)

**Question** : Calculate the sum of all numbers from 1 to a given number.

# Explanation

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

sum\_result = sum(range(1, num + 1))

print("Sum:", sum\_result)

**Question** : Write a Python program to check if a given string is a palindrome.

# Explanation

def is\_palindrome(s):

  return s == s[::-1]

string = input("Enter a string: ")

if is\_palindrome(string):

  print("Palindrome")

else:

  print("Not a palindrome")

**Question**: Count the number of vowels in a given string.

# Explanation

def count\_vowels(s):

  vowels = "aeiouAEIOU"

  return sum(1 for char in s if char in vowels)

string = input("Enter a string: ")

print("Number of vowels:", count\_vowels(string))

**Question** : Reverse a given list in-place.

# Explanation

lst = [1, 2, 3, 4, 5]

lst.reverse()

print(lst)

**Question** : Remove duplicates from a list.

# Explanation

lst = [1, 2, 2, 3, 4, 4, 5]

unique\_lst = list(set(lst))

print(unique\_lst)

**Question** : Check if a given number is prime.

Recommended by LinkedIn

[[](https://www.linkedin.com/pulse/python-beauty-language-deiwin-ignacio-monsalve-altamar)](https://www.linkedin.com/pulse/python-beauty-language-deiwin-ignacio-monsalve-altamar)

**[Python a beauty of language](https://www.linkedin.com/pulse/python-beauty-language-deiwin-ignacio-monsalve-altamar)**

[Deiwin Ignacio M.  4 years ago](https://www.linkedin.com/pulse/python-beauty-language-deiwin-ignacio-monsalve-altamar)

[[](https://www.linkedin.com/pulse/python-decision-making-statements-syntax-malini-shukla)](https://www.linkedin.com/pulse/python-decision-making-statements-syntax-malini-shukla)

**[Python Decision Making Statements with Syntax](https://www.linkedin.com/pulse/python-decision-making-statements-syntax-malini-shukla)**

[Malini Shukla  6 years ago](https://www.linkedin.com/pulse/python-decision-making-statements-syntax-malini-shukla)

[[](https://www.linkedin.com/pulse/9-things-know-master-list-comprehensions-python-prashant-srivastava)](https://www.linkedin.com/pulse/9-things-know-master-list-comprehensions-python-prashant-srivastava)

**[9 Things to Know to Master List Comprehensions in…](https://www.linkedin.com/pulse/9-things-know-master-list-comprehensions-python-prashant-srivastava)**

[Prashant Srivastav  2 years ago](https://www.linkedin.com/pulse/9-things-know-master-list-comprehensions-python-prashant-srivastava)

# Explanation

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

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

if is\_prime(num):

  print("Prime")

else:

  print("Not prime")

**Question**: Convert a string to uppercase.

# Explanation

string = "hello world"

upper\_string = string.upper()

print(upper\_string)

**Question**: Calculate the area of a circle with a given radius.

# Explanation

import math

radius = float(input("Enter the radius of the circle: "))

area = math.pi \* radius\*\*2

print("Area of the circle:", area)

**Question** : Replace all occurrences of a character in a string.

# Explanation

string = "hello world"

new\_string = string.replace('l', 'L')

print(new\_string)

**Question**: Write a Python program to find the maximum element in a list.

# Explanation

lst = [10, 25, 7, 30, 15]

max\_element = max(lst)

print("Maximum element:", max\_element)

**Question** : Calculate the square root of a given number.

# Explanation

import math

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

sqrt\_result = math.sqrt(num)

print("Square root:", sqrt\_result)

**Question** : Check if two strings are anagrams.

# Explanation

def are\_anagrams(str1, str2):

  return sorted(str1) == sorted(str2)

string1 = input("Enter the first string: ")

string2 = input("Enter the second string: ")

if are\_anagrams(string1, string2):

  print("Anagrams")

else:

  print("Not anagrams")

**Question** : Check if a list is empty.

# Explanation

lst = []

if not lst:

  print("List is empty")

else:

  print("List is not empty")

**Question** : Calculate the power of a number.

# Explanation

base = float(input("Enter the base: "))

exponent = float(input("Enter the exponent: "))

power\_result = base \*\* exponent

print("Result:", power\_result)

**Question** : Find the length of the longest word in a given sentence.

# Explanation

sentence = input("Enter a sentence: ")

words = sentence.split()

max\_length = max(len(word) for word in words)

print("Length of the longest word:", max\_length)

**Question** : Check if a given number is a perfect square.

# Explanation

def is\_perfect\_square(num):

  return int(num\*\*0.5)\*\*2 == num

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

if is\_perfect\_square(num):

  print("Perfect square")

else:

  print("Not a perfect square")

**Question**: Find the common elements between two lists.

# Explanation

list1 = [1, 2, 3, 4, 5]

list2 = [4, 5, 6, 7, 8]

common\_elements = list(set(list1) & set(list2))

print("Common elements:", common\_elements)

**Question** : Capitalize the first letter of each word in a sentence.

# Explanation

sentence = "hello world"

capitalized\_sentence = sentence.title()

print(capitalized\_sentence)

**Question**: Print the Fibonacci sequence up to a given number of terms.

# Explanation

def fibonacci(n):

  fib\_seq = [0, 1]

  while len(fib\_seq) < n:

    fib\_seq.append(fib\_seq[-1] + fib\_seq[-2])

  return fib\_seq

num\_terms = int(input("Enter the number of terms: "))

fibonacci\_seq = fibonacci(num\_terms)

print("Fibonacci sequence:", fibonacci\_seq)