**#1. Write a Python program to calculate the area of a rectangle given its length and width.**

**Code:**

def calculate\_rectangle\_area(length, width):

 return length \* width

def main():

  length = float(input("Enter the length of the rectangle: "))

  width = float(input("Enter the width of the rectangle: "))

  area = calculate\_rectangle\_area(length, width)

  print("The area of the rectangle is:", area)

if \_\_name\_\_ =="\_\_main\_\_":

  main()

**Output:**

Enter the length of the rectangle: 12

Enter the width of the rectangle: 12

The area of the rectangle is: 144.0

**#2. Write a program to convert miles to kilometers**

**Code:**

def miles\_to\_kilometers(miles):

    kilometers = miles \* 1.60934

    return kilometers

def main():

    miles = float(input("Enter the distance in miles: "))

    kilometers = miles\_to\_kilometers(miles)

    print(f"{miles} miles is equal to {kilometers} kilometers.")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

    # Define numbers in a list

    numbers = [1,2,3,4,5]

    largest = 0

    for num in numbers:

      if num > largest:

        largest = num

        second\_largest = largest

      elif num > second\_largest:

        second\_largest = num

**Output:**

Enter the distance in miles: 13

13.0 miles is equal to 20.92142 kilometers.

**#3. Write a function to check if a given string is a palindrome**

**Code:**

def is\_palindrome(s):

  # Convert the string to lowercase and remove non-alphanumeric characters

  s = ''.join(char.lower() for char in s if char.isalnum())

  # Check if the string is equal to its reverse

  return s == s[::-1]

# Test the function

def main():

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

  if is\_palindrome(test\_string):

    print("The string is a palindrome.")

  else:

    print("The string is not a palindrome.")

if \_\_name\_\_ == "\_\_main\_\_":

  main()

**Output:**

Enter a string: radar

The string is a palindrome.

**#4. Write a Python program to find the second largest element in a list**

**Code:**

def second\_largest(numbers):

    if len(numbers) < 2:

        return "List must have at least two elements"

    largest = second\_largest = float('-inf')

    for num in numbers:

        if num > largest:

            second\_largest = largest

            largest = num

        elif num > second\_largest and num != largest:

            second\_largest = num

    if second\_largest == float('-inf'):

        return "There is no second largest element"

    else:

        return second\_largest

# Test the function

def main():

    nums = [int(x) for x in input("Enter elements of the list separated by space: ").split()]

    result = second\_largest(nums)

    print("The second largest element in the list is:", result)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Output:**

Enter elements of the list separated by space: 1 3 4 6 8

The second largest element in the list is: 6

**#5. Explain what indentation means in Python**

**Code:**

x=15

if x > 5:

  print("x is greater than 5")

  print("This line is also part of the if block")

print("This line is not indented, so it's not part of the if block")

**Output:**

x is greater than 5

This line is also part of the if block

This line is not indented, so it's not part of the if block

**#6. Write a program to perform set difference operation**

**Code:**

def set\_difference\_using\_operator(set1, set2):

    return set1 - set2

def set\_difference\_using\_method(set1, set2):

    return set1.difference(set2)

# Test the functions

def main():

    set1 = {9,8,5,4,3,2}

    set2 = {2,4,7,5,8,9}

    difference\_operator = set\_difference\_using\_operator(set1, set2)

    difference\_method = set\_difference\_using\_method(set1, set2)

    print("Set difference using operator:", difference\_operator)

    print("Set difference using method:", difference\_method)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Output:**

Set difference using operator: {3}

Set difference using method: {3}

**#7. Write a Python program to print numbers from 1 to 10 using a while loop**

**Code:**

def print\_numbers():

  num = 1

  while num <= 5:

    print(num)

    num += 1

# Test the function

def main():

  print("Numbers from 1 to 10:")

  print\_numbers()

if \_\_name\_\_ == "\_\_main\_\_":

  main()

**Output:**

Numbers from 1 to 10:

1

2

3

4

5

**#8. Write a program to calculate the factorial of a number using a while loop**

**Code:**

def factorial(n):

    if n < 0:

        return "Factorial is not defined for negative numbers"

    elif n == 0:

        return 1

    else:

        result = 1

        while n > 0:

            result \*= n

            n -= 1

        return result

# Test the function

def main():

    num = int(input("Enter a number to calculate its factorial: "))

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

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Output:**

Enter a number to calculate its factorial: 15

Factorial of 15 is 1307674368000

**#9. Write a Python program to check if a number is positive, negative, or zero using if-elif-else**

**Code:**

def check\_number(num):

    if num > 0:

        print("The number is positive.")

    elif num < 0:

        print("The number is negative.")

    else:

        print("The number is zero.")

# Test the function

def main():

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

    check\_number(num)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Output:**

Enter a number: -23

The number is negative.

**#10. Write a program to determine the largest among three numbers using conditional**

**Code:**

def find\_largest(num1, num2, num3):

    if num1 >= num2 and num1 >= num3:

        return num1

    elif num2 >= num1 and num2 >= num3:

        return num2

    else:

        return num3

# Test the function

def main():

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

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

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

    largest = find\_largest(num1, num2, num3)

    print("The largest number among", num1, ",", num2, ", and", num3, "is", largest)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Output:**

Enter the first number: 25

Enter the second number: 32

Enter the third number: 51

The largest number among 25.0 , 32.0 , and 51.0 is 51.0

**#11. Write a Python program to create a numpy array filled with ones of given shape**

**Code:**

import numpy as np

def create\_ones\_array(shape):

    return np.ones(shape)

# Test the function

def main():

    shape = tuple(map(int, input("Enter the shape of the array (separated by spaces): ").split()))

    ones\_array = create\_ones\_array(shape)

    print("Array of ones with shape", shape, ":\n", ones\_array)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Output:**

Enter the shape of the array (separated by spaces): 3 2

Array of ones with shape (3, 2) :

[[1. 1.]

[1. 1.]

[1. 1.]]

**#12. Write a program to create a 2D numpy array initialized with random integers**

**Code:**

import numpy as np

def create\_random\_array(rows, cols):

    return np.random.randint(0, 100, size=(rows, cols))

# Test the function

def main():

    rows = int(input("Enter the number of rows: "))

    cols = int(input("Enter the number of columns: "))

    random\_array = create\_random\_array(rows, cols)

    print("Random array:\n", random\_array)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Output:**

Enter the number of rows: 4

Enter the number of columns: 2

Random array:

[[92 0]

[92 21]

[17 77]

[12 27]]

**#13. Write a Python program to generate an array of evenly spaced numbers over a specified**

**Code:**

import numpy as np

def generate\_evenly\_spaced(start, stop, num):

    return np.linspace(start, stop, num)

# Test the function

def main():

    start = float(input("Enter the start value: "))

    stop = float(input("Enter the stop value: "))

    num = int(input("Enter the number of elements: "))

    evenly\_spaced\_array = generate\_evenly\_spaced(start, stop, num)

    print("Array of evenly spaced numbers:\n", evenly\_spaced\_array)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Output:**

Enter the start value: 2

Enter the stop value: 26

Enter the number of elements: 6

Array of evenly spaced numbers:

[ 2. 6.8 11.6 16.4 21.2 26. ]