1.write a python program to calculate the area of a rectangle given its length and width.

Code:

```
def calculate_rectangle_area(length, width):
    area = length * width
    return area
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(f"The area of the rectangle with length {length} and width {width} is: {area}")
```

2.write a program to convert miles to kilometers.

Code:

```
def miles_to_kilometers(miles):
    kilometers = miles * 1.60934
    return kilometers
miles = float(input("Enter the distance in miles: "))
kilometers = miles_to_kilometers(miles)
print(f"{miles} miles is equal to {kilometers} kilometers")
```

3.write a function to check if a given string is a palindrome.

Code:

```
def is_palindrome(s):
    s = s.replace(" ", "").lower()
    return s == s[::-1]
input_string = input("Enter a string to check if it's a palindrome: ")

if is_palindrome(input_string):
    print(f"{input_string} is a palindrome.")
else:
    print(f"{input_string} is not a palindrome.")
```

4.write a python program to find the second largest element in a list.

Code:

```
def second_largest_element(nums):
    if len(nums) < 2:
        return "List should have at least two elements."
    sorted_nums = sorted(set(nums), reverse=True)
  return sorted_nums[1]
input_list = list(map(int, input("Enter a list of numbers separated by space: ").split()))

result = second_largest_element(input_list)

print(f"The second largest element in the list is: {result}")
```

5.Explain what indentation means in python?

Code:

```
for i in range(5):
    print(i)
```

6.write a program to perform set difference operation.

Code:

```
def set_difference(set_a, set_b):
    return set_a - set_b
if __name__ == "__main__":
    set_a = {1, 2, 3, 4, 5}
    set_b = {3, 4, 5, 6, 7}
result = set_difference(set_a, set_b)
print(f"Set A: {set_a}")
print(f"Set B: {set_b}")
print(f"Set A - B (Difference): {result}")
```

7.write a python program to print numbers from 1 to 10 using a while loop.

Code:

```
number = 1
while number <= 10:
    print(number)
    number += 1
```

8.write a program to calculate the factorial of a number using a while loop.

Code:

```
def calculate_factorial(number):
    result = 1
    while number > 0:
        result *= number
        number -= 1
    return result
if __name__ == "__main__":
    input_number = 5
    factorial_result = calculate_factorial(input_number)
```

```
    print(f"The factorial of {input_number} is:
{factorial_result}")
```

9.write a python program to check if a number is positive ,negative ,or zero using if-elif-else statements.

Code:

```
def check_number_sign(number):
    if number > 0:
        return "Positive"
    elif number < 0:
        return "Negative"
    else:
        return "Zero"
if __name__ == "__main__":
    input_number = float(input("Enter a number: "))
    sign_result = check_number_sign(input_number)
    print(f"The number {input_number} is
{sign_result}.")
```

10.write a program to determine the largest among three numbers using conditional statements.

Code:

```
def find_largest_number(num1, num2, num3):
    if num1 >= num2 and num1 >= num3:
        return num1
    elif num2 >= num1 and num2 >= num3:
        return num2
    else:
        return num3
if __name__ == "__main__":
    num1 = float(input("Enter the first number: "))
    num2 = float(input("Enter the second number: "))
    num3 = float(input("Enter the third number: "))
    largest_number = find_largest_number(num1, num2, num3)
    print(f"The largest number among {num1}, {num2}, and {num3} is: {largest_number}")
```

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):
    ones_array = np.ones(shape)
    return ones_array
if __name__ == "__main__":
    array_shape = tuple(map(int, input("Enter the shape of the array (comma-separated values): ").split(',')))
    ones_array = create_ones_array(array_shape)


    print(f"Array of ones with shape {array_shape}:\n{ones_array}")
```

12.write a program to create a 2D numpy array initialized with random integers.

Code:

```
import numpy as np

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

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

low = int(input("Enter the lower bound (default: 0): ") or 0)

high = int(input("Enter the upper bound (default: 100): ") or 100)

array = np.random.randint(low, high + 1, size=(rows, cols))

print(array)
```

13.write a python program to generate an array of evenly spaced numbers over a specified range using linspace.

Code:

```
import numpy as np

def generate_linspace_array(start, stop, num):
    linspace_array = np.linspace(start, stop, num)
    return linspace_array
```

```
if __name__ == "__main__":
    start_value = float(input("Enter the start value: "))
    stop_value = float(input("Enter the stop value: "))
    num_elements = int(input("Enter the number of elements: "))
    linspace_result = generate_linspace_array(start_value, stop_value, num_elements)
    print(f"Array of {num_elements} evenly spaced numbers from {start_value} to {stop_value}:\n{np.round(linspace_result, 2)}")
```

14.write a python program to generate an array of 10 equally spaced values between 1 and 100 using linspace.

Code:

```
import numpy as np
def generate_linspace_array(start, stop, num):
    linspace_array = np.linspace(start, stop, num)
    return linspace_array
```

```
if __name__ == "__main__":
    linspace_result = generate_linspace_array(1, 100, 10)

    print(f"Array of 10 equally spaced values between 1 and 100:\n{np.round(linspace_result, 2)}")
```

15.write a program to generate an array containing even numbers from 2 to 20 using linspace.

Code:

```
import numpy as np
def generate_even_numbers(start, stop, step):
    even_numbers_array = np.arange(start, stop + 1, step)
    return even_numbers_array
if __name__ == "__main__":
```

```
    even_numbers_result = generate_even_numbers(2,
20, 2)
    print(f"Array of even numbers from 2 to
20:\n{even_numbers_result}")
```

16.write a program to create an array containing numbers from 1 to 10 with a step size of 0.5 using arrange.

Code:

```
import numpy as np
def generate_array_with_step(start, stop, step):
    result_array = np.arange(start, stop + step, step)
    return result_array
if __name__ == "__main__":
```

```
    result_array = generate_array_with_step(1, 10, 0.5)

    print(f"Array from 1 to 10 with a step size of
0.5:\n{result_array}")
```