1.

main.py

```python
def calculate_rectangle_area(length, width):
    area = length * width
    return area

# Taking input from the user
length = float(input("Enter the length of the rectangle: "))
width = float(input("Enter the width of the rectangle: "))

# Calculating the area
area = calculate_rectangle_area(length, width)

# Displaying the result
print(f"The area of the rectangle with length {length} and width {width} is: {area}")
```

Shell

```
Enter the length of the rectangle: 5
Enter the width of the rectangle: 4
The area of the rectangle with length 5.0 and width 4.0 is: 20.0
```

2.

main.py

```python
def miles_to_kilometers(miles):
    kilometers = miles * 1.60934
    return kilometers

# Taking input from the user
miles = float(input("Enter the distance in miles: "))

# Converting miles to kilometers
kilometers = miles_to_kilometers(miles)

# Displaying the result
print(f"{miles} miles is equal to {kilometers:.2f} kilometers")
```

Shell

```
Enter the distance in miles: 8
8.0 miles is equal to 12.87 kilometers
```

3.

main.py

```python
def is_palindrome(s):
    s = s.lower()  # Convert the string to lowercase for case-insensitive comparison
    return s == s[::-1]

# Example usage:
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.")
```

Shell

```
Enter a string to check if it's a palindrome: 454
454 is a palindrome.
```

4.

main.py

```python
def find_second_largest(lst):
    if len(lst) < 2:
        return "List should have at least two elements."

    sorted_list = sorted(set(lst), reverse=True)
    return sorted_list[1]

# Example usage:
numbers = [int(x) for x in input("Enter a list of numbers separated by space: ").split()]
second_largest = find_second_largest(numbers)
print(f"The second largest element in the list is: {second_largest}")
```

Shell

```
Enter a list of numbers separated by space: 4 45 6 5
The second largest element in the list is: 6
```

5.

In Python, indentation is used to define the block of code. Unlike many other programming languages that use braces {}, Python uses indentation to indicate the scope of control structures,

functions, classes, and other code blocks. The standard convention is to use four spaces for each level of indentation. Incorrect indentation can lead to syntax errors or unexpected behavior.

6.

main.py

```python
def set_difference_operation(set1, set2):
    difference = set1 - set2
    return difference

# Example usage:
set_a = {1, 2, 3, 4, 5}
set_b = {3, 4, 5, 6, 7}
result = set_difference_operation(set_a, set_b)
print(f"The set difference of {set_a} and {set_b} is: {result}")
```

Shell

```
The set difference of {1, 2, 3, 4, 5} and {3, 4, 5, 6, 7} is: {1, 2}
```

7.

main.py

```python
num = 1
while num <= 10:
    print(num)
    num += 1
```

Shell

```
1
2
3
4
5
6
7
8
9
10
```

8.

main.py

```python
def calculate_factorial(n):
    result = 1
    while n > 0:
        result *= n
        n -= 1
    return result

# Example usage:
number = int(input("Enter a number to calculate its factorial: "))
factorial = calculate_factorial(number)
print(f"The factorial of {number} is: {factorial}")
```

Shell

```
Enter a number to calculate its factorial: 54
The factorial of 54 is:
    23084369733924138047209274268302758108327856457180794113228800000000000
```

9.

main.py

```python
number = float(input("Enter a number: "))

if number > 0:
    print("The number is positive.")
elif number < 0:
    print("The number is negative.")
else:
    print("The number is zero.")
```

Shell

```
Enter a number: 54
The number is positive.
```

10.

main.py

```python
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
num3 = float(input("Enter the third number: "))

if num1 >= num2 and num1 >= num3:
    largest = num1
elif num2 >= num1 and num2 >= num3:
    largest = num2
else:
    largest = num3

print(f"The largest among {num1}, {num2}, and {num3} is: {largest}")
```

Shell

```
Enter the first number: 54
Enter the second number: 78
Enter the third number: 88
The largest among 54.0, 78.0, and 88.0 is: 88.0
>
```

11.

main.py

```python
import numpy as np

shape = tuple(map(int, input("Enter the shape of the array (comma-separated values): ").split(',')))
ones_array = np.ones(shape)
print("Numpy array filled with ones:")
print(ones_array)
```

Shell

```
Enter the shape of the array (comma-separated values): 87
Numpy array filled with ones:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
>
```

12.

main.py

```python
import numpy as np

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

random_array = np.random.randint(1, 100, size=(rows, cols))
print("2D Numpy array initialized with random integers:")
print(random_array)
```

Shell

```
Enter the number of rows: 54
Enter the number of columns: 88
2D Numpy array initialized with random integers:
[[46 53 91 ... 74 76 41]
 [24 13 34 ... 40 80 48]
 [ 3  6 93 ... 39 97 43]
 ...
 [69 69 71 ... 66 78 91]
 [46 87 64 ... 72 79  5]
 [77 78  3 ... 25 24 10]]
>
```

13.

main.py

```python
import numpy as np

start = float(input("Enter the start of the range: "))
end = float(input("Enter the end of the range: "))
num_points = int(input("Enter the number of points: "))

generated_array = np.linspace(start, end, num_points)
print("Generated array using linspace:")
print(generated_array)
```

Shell

```
Enter the start of the range: 7
Enter the end of the range: 8
Enter the number of points: 4
Generated array using linspace:
[7.         7.33333333 7.66666667 8.        ]
>
```

14.

main.py

```python
import numpy as np

equally_spaced_array = np.linspace(1, 100, 10)
print("Array of 10 equally spaced values between 1 and 100:")
print(equally_spaced_array)
```

Shell

```
Array of 10 equally spaced values between 1 and 100:
[  1.  12.  23.  34.  45.  56.  67.  78.  89. 100.]
>
```

15.

main.py

```python
import numpy as np

even_array = np.arange(2, 21, 2)
print("Array containing even numbers from 2 to 20:")
print(even_array)
```

Shell

```
Array containing even numbers from 2 to 20:
[ 2  4  6  8 10 12 14 16 18 20]
>
```

16.

main.py

```python
import numpy as np

array_with_step = np.arange(1, 11, 0.5)
print("Array containing numbers from 1 to 10 with a step size of 0.5:")
print(array_with_step)
```

Shell

```
Array containing numbers from 1 to 10 with a step size of 0.5:
[ 1.   1.5  2.   2.5  3.   3.5  4.   4.5  5.   5.5  6.   6.5  7.   7.5
  8.   8.5  9.   9.5 10.  10.5]
```