```python
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory
```

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

```python
def areaofRectangle(length, breadth):
    return length*breadth
print("The area of Rectangle is : " + str(areaofRectangle(10,4)))
```

```
The area of Rectangle is : 40
```

2) program to convert miles to kilometers

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

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

```
Enter distance in miles:  1

1.0 miles is equal to 1.60934 kilometers
```

3) Function to check if a given string is a palindrome:

```python
def is_palindrome(s):
    return s == s[::-1]

string = input("Enter a string: ")
if is_palindrome(string):
    print("Palindrome")
else:
    print("Not a palindrome")
```

```
Enter a string:  sanas
```

```
Palindrome
```

4) program to find the second largest element in a list:

```
def second_largest(numbers):
    unique_numbers = set(numbers)
    unique_numbers.remove(max(unique_numbers))
    return max(unique_numbers)

numbers = [int(x) for x in input("Enter numbers separated by space: 
").split()]
print("Second largest element:", second_largest(numbers))

Enter numbers separated by space:  1  2 9 8 7 6 0 8 7

Second largest element: 8
```

5) Indentation in Python

Indentation in Python refers to the whitespace at the beginning of a line of code. It is used to define the structure and hierarchy of code blocks. In Python, indentation is used to delimit blocks of code, such as those in loops, conditional statements, function definitions, and classes. It is crucial for code readability and is a fundamental aspect of Python's syntax.

6)Program to perform set difference operation:

```
set1 = {1, 2, 3, 4, 5}
set2 = {3, 4, 5, 6, 7}
difference = set1 - set2
print("Set difference:", difference)

Set difference: {1, 2}
```

7) Python program to print numbers from 1 to 10 using a while loop:

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

1
2
3
4
5
6
7
8
```

```
9
10
```

8) Program to calculate the factorial of a number using a while loop:

```
def factorial(n):
    result = 1
    while n > 1:
        result *= n
        n -= 1
    return result

number = int(input("Enter a number: "))
print("Factorial:", factorial(number))

Enter a number:  6

Factorial: 720
```

9)Python program to check if a number is positive, negative, or zero using if-elif-else statements:

```
number = float(input("Enter a number: "))
if number > 0:
    print("Positive")
elif number < 0:
    print("Negative")
else:
    print("Zero")

Enter a number:  96

Positive
```

10) Program to determine the largest among three numbers using conditional statements:

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

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

print("Largest number:", largest)
```

```
Enter first number:  9
Enter second number:  5
Enter third number:  7

Largest number: 9.0
```

11) Python program to create a numpy array filled with ones of given shape:

```python
import numpy as np

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

```
Enter the shape of the array (comma-separated):  9

Array filled with ones:
[1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

12) Program to create a 2D numpy array initialized with random integers:

```python
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 for random integers: "))
high = int(input("Enter the upper bound for random integers: "))

random_array = np.random.randint(low, high, size=(rows, cols))
print("2D Array with random integers:")
print(random_array)
```

```
Enter the number of rows:  5
Enter the number of columns:  2
Enter the lower bound for random integers:  1
Enter the upper bound for random integers:  50

2D Array with random integers:
[[12 20]
 [45 10]
 [ 6 48]
 [21 39]
 [25 27]]
```

13) Python program to generate an array of evenly spaced numbers over a specified range using linspace:

```
import numpy as np

start = float(input("Enter the start of the range: "))
stop = float(input("Enter the end of the range: "))
num_values = int(input("Enter the number of values: "))

linspace_array = np.linspace(start, stop, num_values)
print("Array of evenly spaced numbers:")
print(linspace_array)
```

```
Enter the start of the range:  1
Enter the end of the range:  50
Enter the number of values:  60

Array of evenly spaced numbers:
[ 1.          1.83050847  2.66101695  3.49152542  4.3220339
5.15254237
  5.98305085  6.81355932  7.6440678   8.47457627  9.30508475
10.13559322
 10.96610169 11.79661017 12.62711864 13.45762712 14.28813559
15.11864407
 15.94915254 16.77966102 17.61016949 18.44067797 19.27118644
20.10169492
 20.93220339 21.76271186 22.59322034 23.42372881 24.25423729
25.08474576
 25.91525424 26.74576271 27.57627119 28.40677966 29.23728814
30.06779661
 30.89830508 31.72881356 32.55932203 33.38983051 34.22033898
35.05084746
 35.88135593 36.71186441 37.54237288 38.37288136 39.20338983
40.03389831
 40.86440678 41.69491525 42.52542373 43.3559322  44.18644068
45.01694915
 45.84745763 46.6779661  47.50847458 48.33898305 49.16949153 50.
]
```

14) Program to generate an array of 10 equally spaced values between 1 and 100 using linspace:

```
import numpy as np

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

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

15) Python program to create an array containing even numbers from 2 to 20 using arange:

```
import numpy as np

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

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

16) Program to create an array containing numbers from 1 to 10 with a step size of 0.5 using arange:

```
import numpy as np

array_with_step = np.arange(1, 10.5, 0.5)
print("Array containing numbers from 1 to 10 with a step size of 0.5:")
print(array_with_step)
```

```
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. ]
```