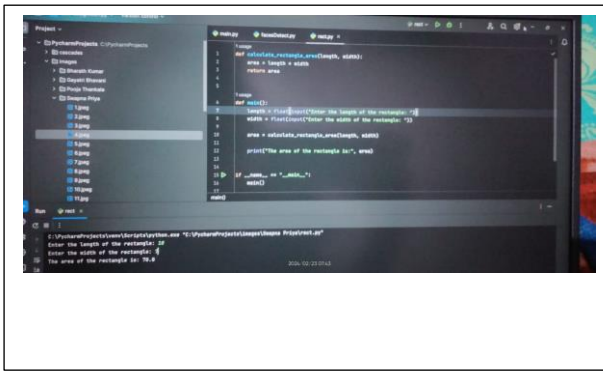
ASSIGNMENT -1

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

Ans: def calculate_rectangle_area (length, width):

```
    area = length * width
    return area


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:

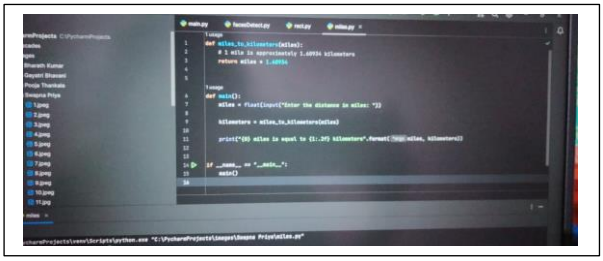
length of rectangle :10

length of width:7

area of the rectangle is:70

2.write a program to convert miles to kilometers?

```
def miles_to_kilometers(miles):
    # 1 mile is approximately 1.60934 kilometers
    return miles * 1.60934


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

    kilometers = miles_to_kilometers(miles)

    print("{0} miles is equal to {1:.2f} kilometers".format(miles, kilometers))

if __name__ == "__main__":
    main()
```



ASSIGNMENT -1

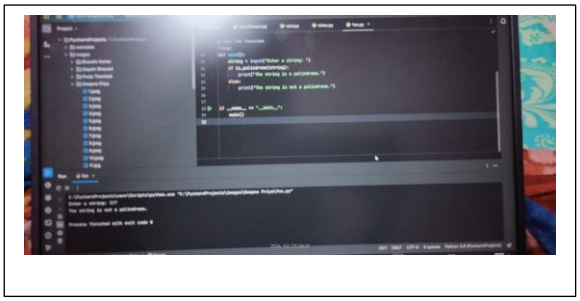
output:

The distance in the miles :450

The 450.0 mile is equal to 724.20 kilometers

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

Ans:
```
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():
    string = input("Enter a string: ")
    if is_palindrome(string):
        print("The string is a palindrome.")
    else:
        print("The string is not a palindrome.")

if __name__ == "__main__":
    main()
```



output:

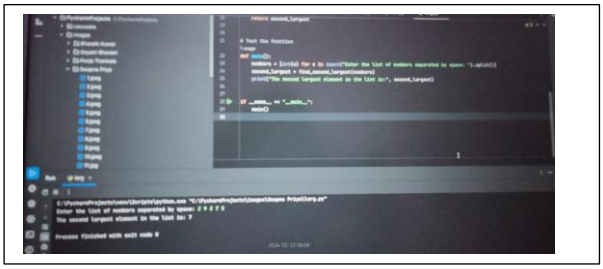
Enter a string:426

The string is not a palindrome

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

Ans:
```
def find_second_largest(numbers):
    # Check if the list has at least two elements
```

ASSIGNMENT -1



```
    if len(numbers) < 2:
        return "List must have at least two elements"

    # Initialize the first and second largest elements
    largest = second_largest = float('-inf')

    # Iterate through the list
    for num in numbers:
        # Update the largest and second largest elements
        if num > largest:
            second_largest = largest
            largest = num
        elif num > second_largest and num != largest:
            second_largest = num

    return second_largest

# Test the function
def main():
    numbers = [int(x) for x in input("Enter the list of numbers separated by space: ").split()]
    second_largest = find_second_largest(numbers)
    print("The second largest element in the list is:", second_largest)

if __name__ == "__main__":
    main()
```

output:

enter the list of numbers separated by space: 2 9 3 7 5

second largest number is :7

5.explain what is indentation means in python?

Ans: Indentation in Python refers to the spaces or tabs that are used at the beginning of a line to define a block of code. In Python, indentation is not just for readability; it is part of the syntax and is used to indicate the structure and hierarchy of the code.

ASSIGNMENT -1

1. **Defining Blocks**: Indentation is used to define blocks of code such as loops, conditional statements, function definitions, and class definitions. Blocks are delineated by indentation levels rather than curly braces or keywords.

2. **Consistency**: Python requires consistent indentation throughout the code. All statements within the same block must be indented to the same level.

3. **Scope**: Indentation determines the scope of variables and statements. Statements with the same level of indentation belong to the same block and have the same scope.

4. **Readability**: Proper indentation enhances code readability by visually showing the structure of the code.

For example, in a conditional statement in Python, the block of code that executes when the condition is true is indented:

6.write a program to perform set difference operation?

Ans:
```
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 = {1, 2, 3, 4, 5}
    set2 = {3, 4, 5, 6, 7}

    # Using operator -
    difference_operator = set_difference_using_operator(set1, set2)
    print("Difference using operator -:", difference_operator)
```

ASSIGNMENT -1

```
    # Using method difference()
    difference_method = set_difference_using_method(set1, set2)
    print("Difference using method difference():", difference_method)

if __name__ == "__main__":
    main()
```

output:

Difference using operator -: {1, 2}

Difference using method difference(): {1, 2}

7.write a python program that prints numbers from 1 to 10 using a while loop?

Ans:

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

# Call the function to print numbers
print_numbers()
```

output:

1

2

3

4

5

6

7

8

9

ASSIGNMENT -1

10

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

Ans:

```
def factorial(n):
    # Initialize the result variable to store the factorial
    result = 1

    # Check if n is non-negative
    if n < 0:
        return "Factorial is not defined for negative numbers"

    # Calculate the factorial using a while loop
    while n > 1:
        result *= n
        n -= 1

    return result

# Test the function
def main():
    number = int(input("Enter a number to calculate its factorial: "))
    fact = factorial(number)
    print("The factorial of", number, "is:", fact)

if __name__ == "__main__":
    main()
```

output:

enter a number to calculate its factorial:13

the factorial of 13 is :6227020800

ASSIGNMENT -1

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

Ans:
```
def check_number(num):
    if num > 0:
        print(num, "is positive")
    elif num < 0:
        print(num, "is negative")
    else:
        print(num, "is zero")

# Test the function
def main():
    number = float(input("Enter a number: "))
    check_number(number)

if __name__ == "__main__":
    main()
```

output:

enter the number:9

9.0 is a positive number

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

Ans:
```
def find_largest(num1, num2, num3):
    if num1 >= num2 and num1 >= num3:
        largest = num1
    elif num2 >= num1 and num2 >= num3:
        largest = num2
    else:
        largest = num3
    return largest
```

ASSIGNMENT -1

```
# 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:24

enter the second number:45

enter the third number:52

among three numbers the largest one is:52.0

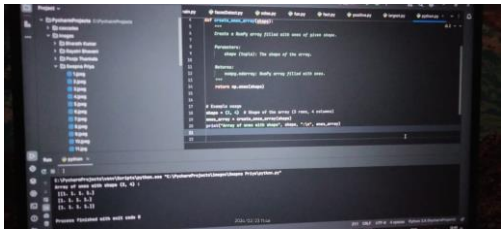
11.write a python program to create a numpy array filled with ones of given shape?

Ans:

```
import numpy as np

def create_ones_array(shape):
    """
    Create a NumPy array filled with ones of given shape.

    Parameters:
        shape (tuple): The shape of the array.

    Returns:
        numpy.ndarray: NumPy array filled with ones.
    """
```

ASSIGNMENT -1

```
    return np.ones(shape)


# Example usage
shape = (3, 4)  # Shape of the array (3 rows, 4 columns)
ones_array = create_ones_array(shape)
print("Array of ones with shape", shape, ":\n", ones_array)
```



output:

array of ones with shape(3,4):

[[1, 1, 1, 1,]

[1, 1, 1, 1,]

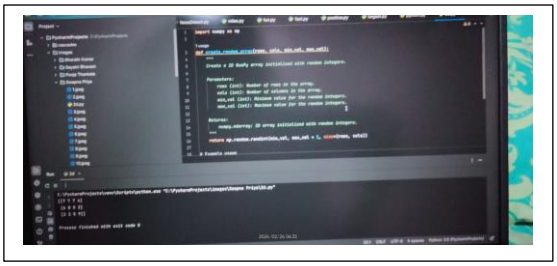
[1, 1, 1, 1,]]

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

Ans: import numpy as np

```
def create_random_array(rows, cols, min_val, max_val):
    """
    Create a 2D NumPy array initialized with random integers.

    Parameters:
        rows (int): Number of rows in the array.
        cols (int): Number of columns in the array.
        min_val (int): Minimum value for the random integers.
        max_val (int): Maximum value for the random integers.

    Returns:
        numpy.ndarray: 2D array initialized with random integers.
    """
    return np.random.randint(min_val, max_val + 1, size=(rows, cols))
```

ASSIGNMENT -1



```
# Example usage
rows = 3
cols = 4
min_val = 0
max_val = 10
random_array = create_random_array(rows, cols, min_val, max_val)
print(random_array)
```

output:

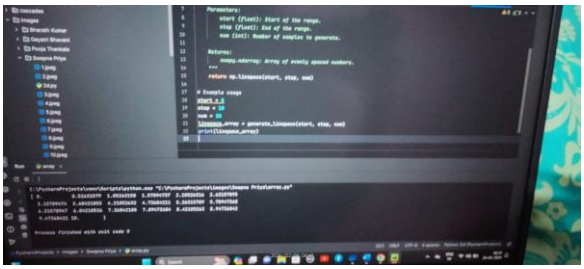
```
[[7 7 7 4]
[6 0 5 3]
[2 2 5 9]]
```

13.write a python program to generate an array of evenly space number over a specified range using linspace ?

Ans:

```
import numpy as np

def generate_linspace(start, stop, num):
    """
    Generate an array of evenly spaced numbers over a specified range using linspace.

    Parameters:
        start (float): Start of the range.
        stop (float): End of the range.
        num (int): Number of samples to generate.

    Returns:
        numpy.ndarray: Array of evenly spaced numbers.
    """
    return np.linspace(start, stop, num)
```

ASSIGNMENT -1



Example usage

start = 0

stop = 10

num = 20

linspace_array = generate_linspace(start, stop, num)

print(linspace_array)

output:

[0. 0.52631579 1.05263158 1.57894737 2.10526316 2.63157895

3.15789474 3.68421053 4.21052632 4.73684211 5.26315789 5.78947368

6.31578947 6.84210526 7.36842105 7.89473684 8.42105263 8.94736842

9.47368421 10.]

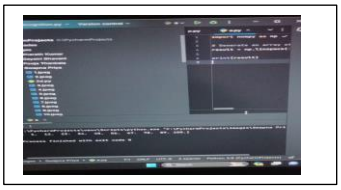
14.write a python program to generate an array of 10 equally spaced values between 1 and 100 using linspace?

Ans:

import numpy as np

Generate an array of 10 equally spaced values between 1 and 100

result = np.linspace(1, 100, 10)



print(result)

output:

[1. 12. 23. 34. 45. 56. 67. 78. 89. 100.]

ASSIGNMENT -1

15.write a python program to create an array containing even number from 2 to 20 using arrange?

Ans:

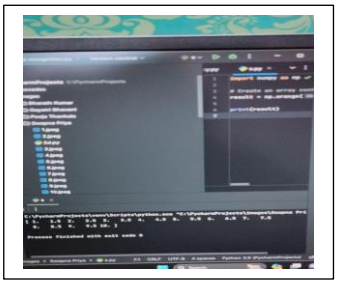
```
import numpy as np

# Create an array containing even numbers from 2 to 20
result = np.arange(2, 21, 2)

print(result)
```

output:

```
[ 2  4  6  8 10 12 14 16 18 20]
```

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

Ans:



```
import numpy as np

# Create an array containing numbers from 1 to 10 with a step size of 0.5
result = np.arange(1, 10.5, 0.5)

print(result)
```

output:

```
[ 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. ]
```