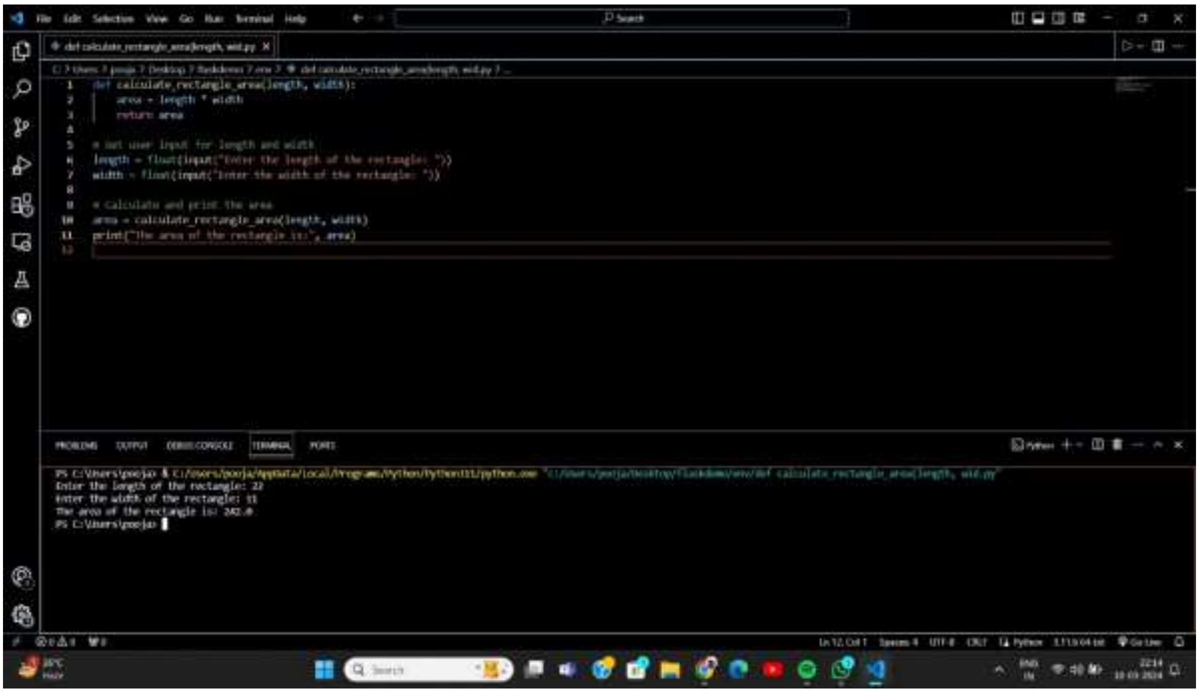
Assignment-1

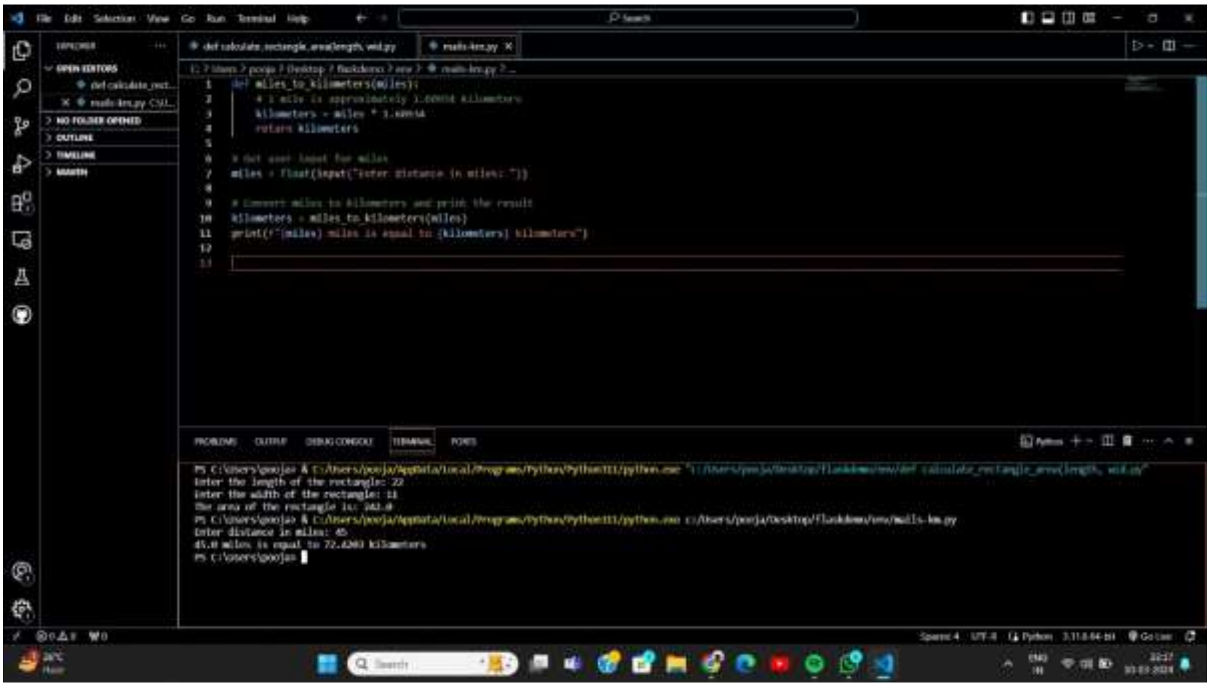
1.write a python program to calculate the area of rectangle given its length and width.



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

# Get user input for length and width
length = float(input("Enter the length of the rectangle: "))
width = float(input("Enter the width of the rectangle: "))

# Calculate and print the area
area = calculate_rectangle_area(length, width)
print("The area of the rectangle is:", area)
```

```
PS C:\Users\poojas> & C:/Users/pooja/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/pooja/Desktop/flaskdemo/env/def calculate_rectangle_area(length, wid.py"
Enter the length of the rectangle: 22
Enter the width of the rectangle: 11
The area of the rectangle is: 242.0
PS C:\Users\poojas>
```

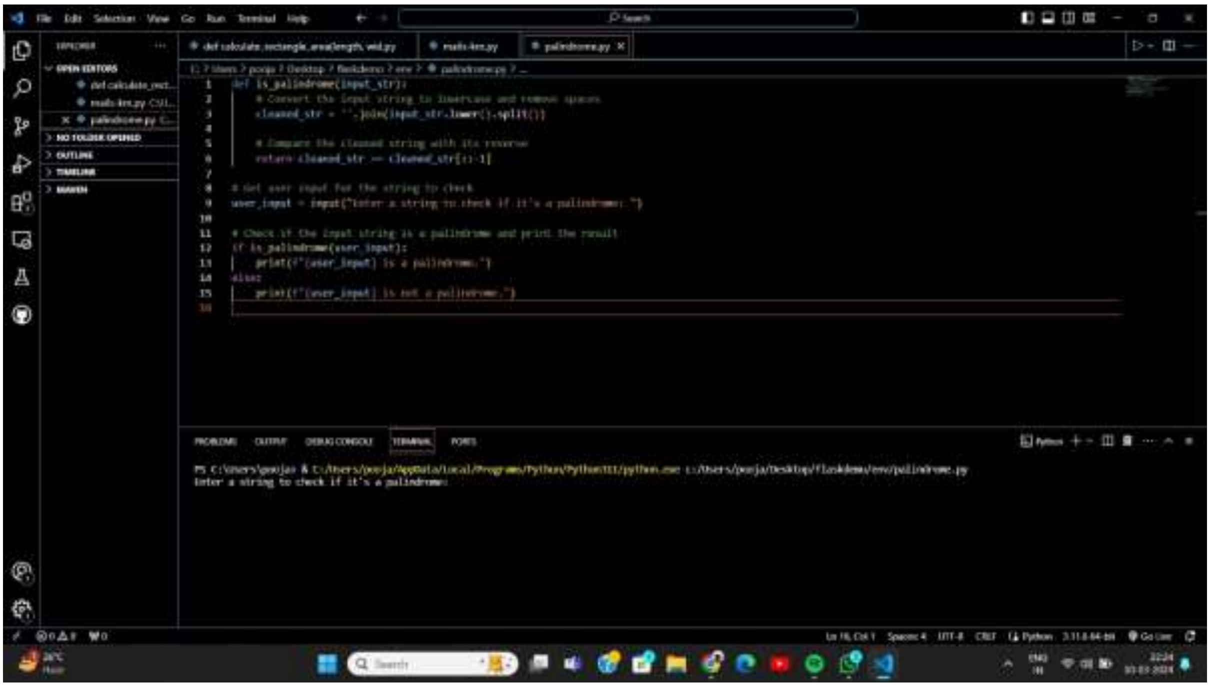
2.write a program to coonvert miles to kilometers



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

# Get user input for miles
miles = float(input("Enter distance in miles: "))

# Convert miles to kilometers and print the result
kilometers = miles_to_kilometers(miles)
print(f"{miles} miles is equal to {kilometers} kilometers")
```

```
PS C:\Users\poojas> & C:/Users/pooja/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/pooja/Desktop/flaskdemo/env/def calculate_rectangle_area(length, wid.py"
Enter the length of the rectangle: 22
Enter the width of the rectangle: 11
The area of the rectangle is: 242.0
PS C:\Users\poojas> & C:/Users/pooja/AppData/Local/Programs/Python/Python311/python.exe c:/Users/pooja/Desktop/flaskdemo/env/mails-km.py
Enter distance in miles: 45
45.0 miles is equal to 72.4203 kilometers
PS C:\Users\poojas>
```

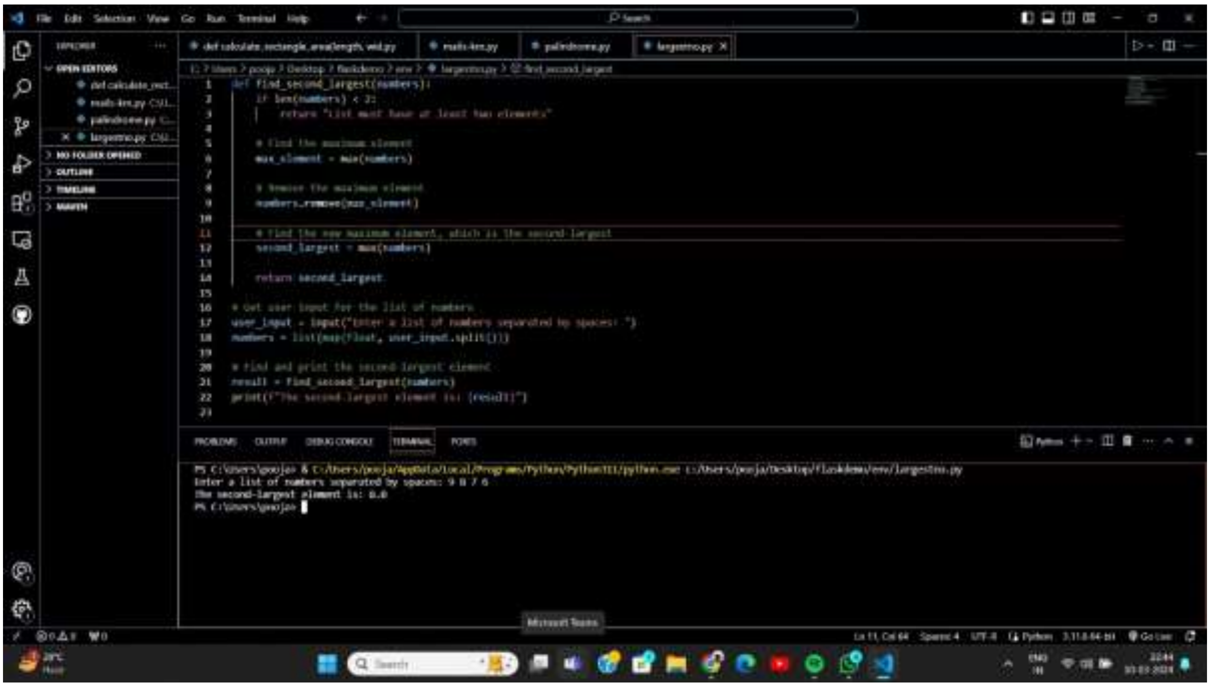
3.write a function to check if a given string is a palindrome


File Edit Selection View Go Run Terminal Help
def calculate_rectangle_area(length, wid.py
maths-4m.py
palindrome.py
C: > Users > pooja > Desktop > flaskdemo > env > palindrome.py > ...
OPEN EDITORS
NO FOLDER OPENED
OUTLINE
TIMELINE
MAVEN
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```python
def is_palindrome(input_str):
    # Convert the input string to lowercase and remove spaces
    cleaned_str = ''.join(input_str.lower().split())

    # Compare the cleaned string with its reverse
    return cleaned_str == cleaned_str[::-1]

# Get user input for the string to check
user_input = input("Enter a string to check if it's a palindrome: ")

# Check if the input string is a palindrome and print the result
if is_palindrome(user_input):
    print(f"{user_input} is a palindrome.")
else:
    print(f"{user_input} is not a palindrome.")

```

```
PS C:\Users\pooja> & C:/Users/pooja/AppData/Local/Programs/Python/Python311/python.exe c:/Users/pooja/Desktop/flaskdemo/env/palindrome.py
Enter a string to check if it's a palindrome:
```

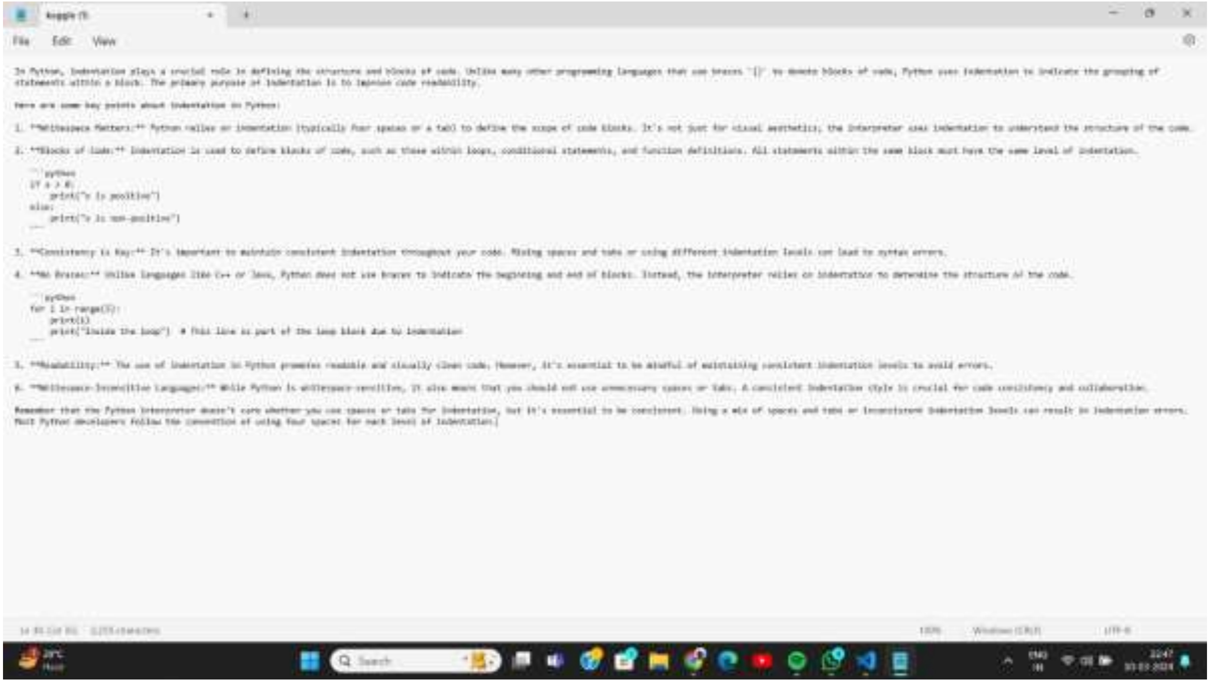
4.write a python program to find the second largest element in a list


File Edit Selection View Go Run Terminal Help
def calculate_rectangle_area(length, wid.py
maths-4m.py
palindrome.py
largestno.py
C: > Users > pooja > Desktop > flaskdemo > env > largestno.py > find_second_largest
OPEN EDITORS
NO FOLDER OPENED
OUTLINE
TIMELINE
MAVEN
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Microsoft Teams

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

    # Find the maximum element
    max_element = max(numbers)

    # Remove the maximum element
    numbers.remove(max_element)

    # Find the new maximum element, which is the second largest
    second_largest = max(numbers)

    return second_largest

# Get user input for the list of numbers
user_input = input("Enter a list of numbers separated by spaces: ")
numbers = list(map(float, user_input.split()))

# Find and print the second largest element
result = find_second_largest(numbers)
print(f"The second largest element is: {result}")

```

```
PS C:\Users\pooja> & C:/Users/pooja/AppData/Local/Programs/Python/Python311/python.exe c:/Users/pooja/Desktop/flaskdemo/env/largestno.py
Enter a list of numbers separated by spaces: 9 8 7 6
The second-largest element is: 8.0
PS C:\Users\pooja> 
```

5.explain what indentation means in python



In Python, indentation plays a crucial role in defining the structure and blocks of code. Unlike many other programming languages that use braces `{}` to denote blocks of code, Python uses indentation to indicate the grouping of statements within a block. The primary purpose of indentation is to improve code readability.

Here are some key points about indentation in Python:

1. **Whitespace Matters:** Python relies on indentation (typically four spaces or a tab) to define the scope of code blocks. It's not just for visual aesthetics; the interpreter uses indentation to understand the structure of the code.

2. **Blocks of Code:** Indentation is used to define blocks of code, such as those within loops, conditional statements, and function definitions. All statements within the same block must have the same level of indentation.

~~~
   ```python
   if x > 0:
       print("x is positive")
   else:
       print("x is non-positive")
   ```
~~~

3. **Consistency is Key:** It's important to maintain consistent indentation throughout your code. Mixing spaces and tabs or using different indentation levels can lead to syntax errors.

4. **No Braces:** Unlike languages like C++ or Java, Python does not use braces to indicate the beginning and end of blocks. Instead, the interpreter relies on indentation to determine the structure of the code.

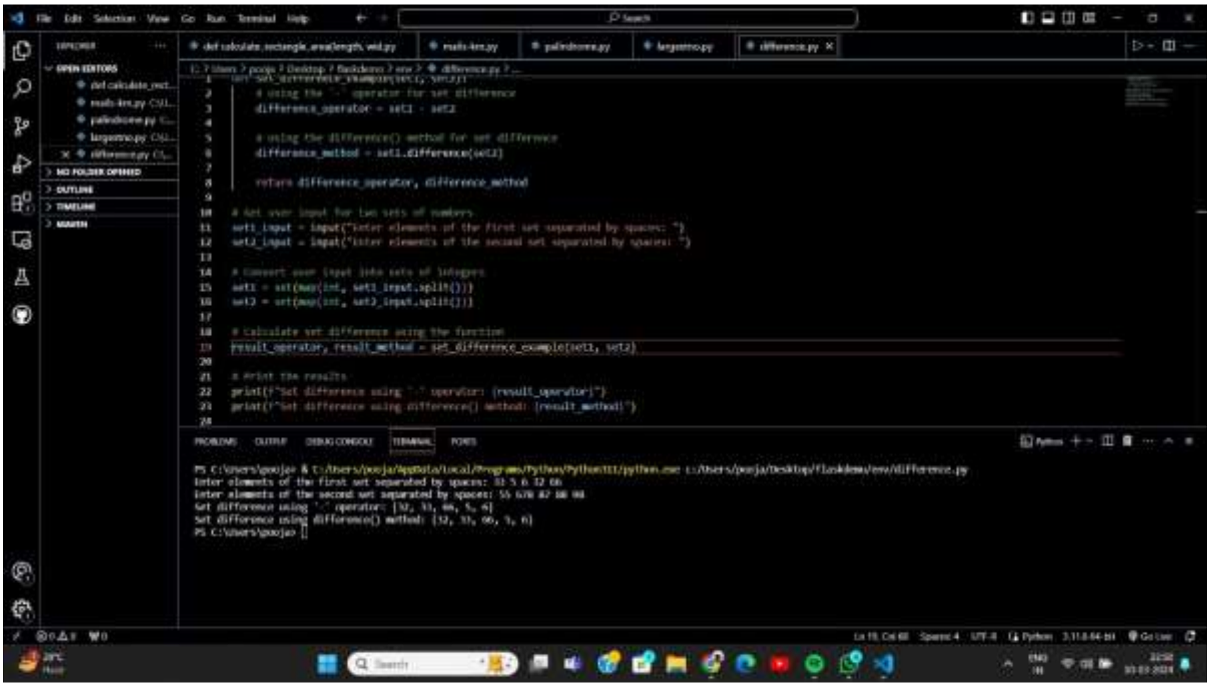
~~~
   ```python
   for i in range(5):
       print(i)
       print("Inside the loop")  # This line is part of the loop block due to indentation
   ```
~~~

5. **Readability:** The use of indentation in Python promotes readable and visually clean code. However, it's essential to be mindful of maintaining consistent indentation levels to avoid errors.

6. **Whitespace-Insensitive Languages:** While Python is whitespace-sensitive, it also means that you should not use unnecessary spaces or tabs. A consistent indentation style is crucial for code consistency and collaboration.

Remember that the Python interpreter doesn't care whether you use spaces or tabs for indentation, but it's essential to be consistent. Using a mix of spaces and tabs or inconsistent indentation levels can result in indentation errors. Most Python developers follow the convention of using four spaces for each level of indentation.

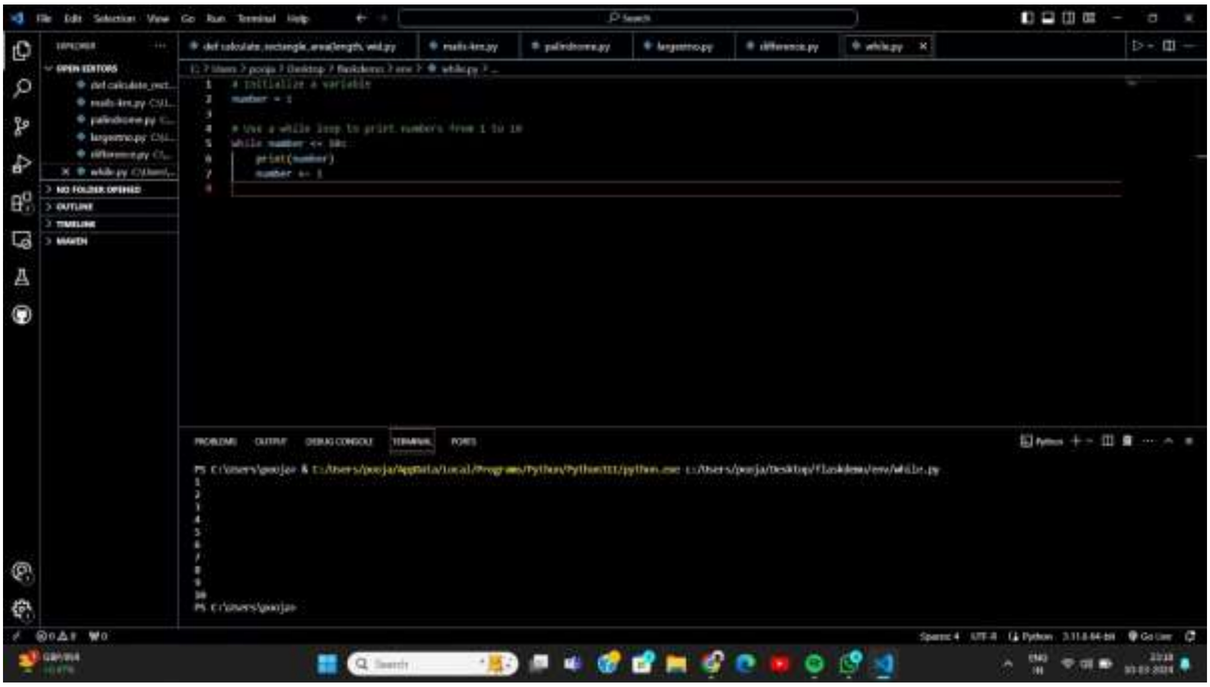
6.write program to perform set difference operation



```
def set_difference_example(set1, set2):
    # using the '-' operator for set difference
    difference_operator = set1 - set2

    # using the difference() method for set difference
    difference_method = set1.difference(set2)

    return difference_operator, difference_method

# Get user input for two sets of numbers
set1_input = input("Enter elements of the first set separated by spaces: ")
set2_input = input("Enter elements of the second set separated by spaces: ")

# Convert user input into sets of integers
set1 = set(map(int, set1_input.split()))
set2 = set(map(int, set2_input.split()))

# Calculate set difference using the function
result_operator, result_method = set_difference_example(set1, set2)

# Print the results
print(f"Set difference using '-' operator: {result_operator}")
print(f"Set difference using difference() method: {result_method}")
```

```
PS C:\Users\poojaa> & C:/Users/pooja/AppData/Local/Programs/Python/Python311/python.exe c:/Users/pooja/Desktop/flaskdemo/env/difference.py
Enter elements of the first set separated by spaces: 33 5 6 32 66
Enter elements of the second set separated by spaces: 55 678 87 88 98
Set difference using '-' operator: {32, 33, 66, 5, 6}
Set difference using difference() method: {32, 33, 66, 5, 6}
PS C:\Users\poojaa> 
```

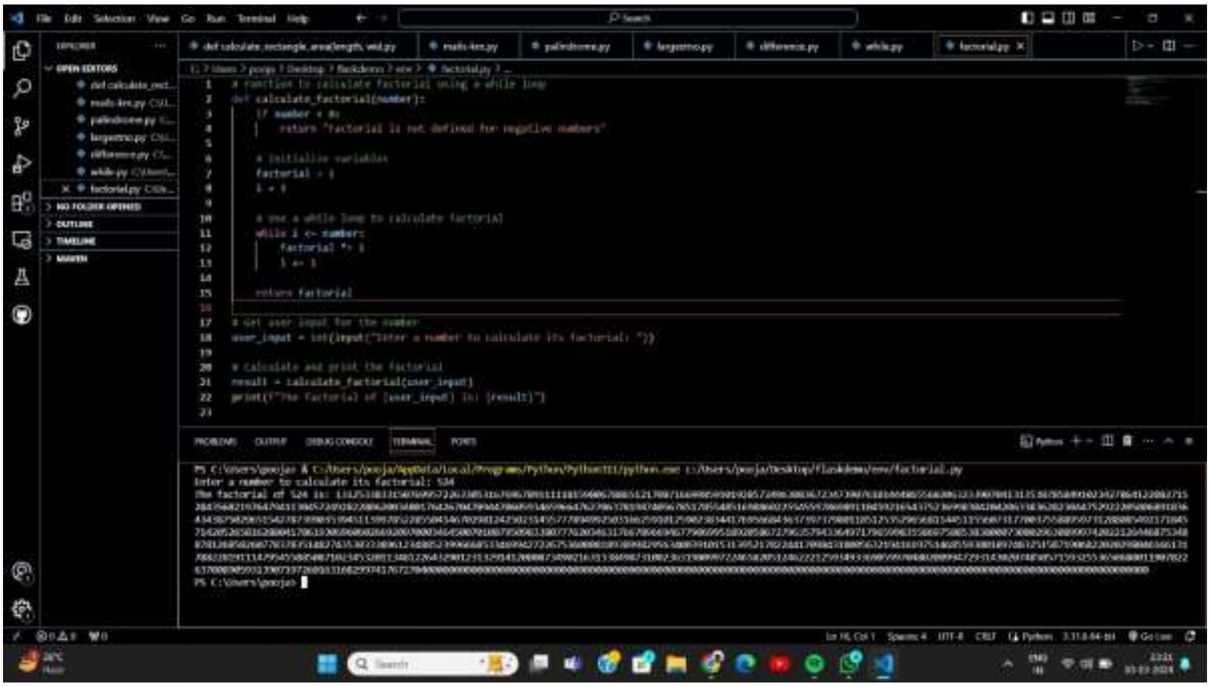
7.write a python program to print numbers from 1 to 10 using while loop



```python
# Initialize a variable
number = 1

# Use a while loop to print numbers from 1 to 10
while number <= 10:
    print(number)
    number += 1
```

```
PS C:\Users\pooja> & C:/Users/pooja/AppData/Local/Programs/Python/Python311/python.exe c:/Users/pooja/Desktop/flaskdemo/env/while.py
1
2
3
4
5
6
7
8
9
10
PS C:\Users\pooja>
```

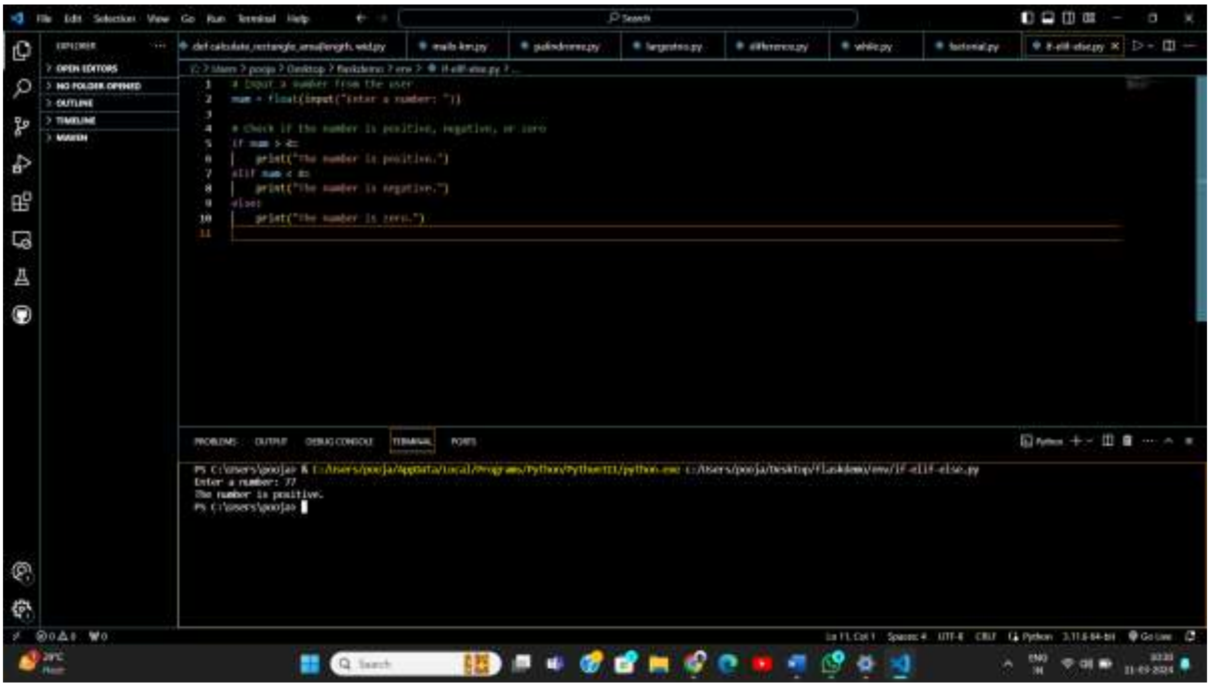
8.write a program to calculate the factorial of a number using while loop



```python
# Function to calculate factorial using a while loop
def calculate_factorial(number):
    if number < 0:
        return "Factorial is not defined for negative numbers"

    # Initialize variables
    factorial = 1
    i = 1

    # Use a while loop to calculate factorial
    while i <= number:
        factorial *= i
        i += 1

    return factorial

# Get user input for the number
user_input = int(input("Enter a number to calculate its factorial: "))

# Calculate and print the factorial
result = calculate_factorial(user_input)
print(f"The factorial of {user_input} is: {result}")
```

```
PS C:\Users\pooja> & C:/Users/pooja/AppData/Local/Programs/Python/Python311/python.exe c:/Users/pooja/Desktop/flaskdemo/env/factorial.py
Enter a number to calculate its factorial: 524
The factorial of 524 is: [illegible]
PS C:\Users\pooja>
```

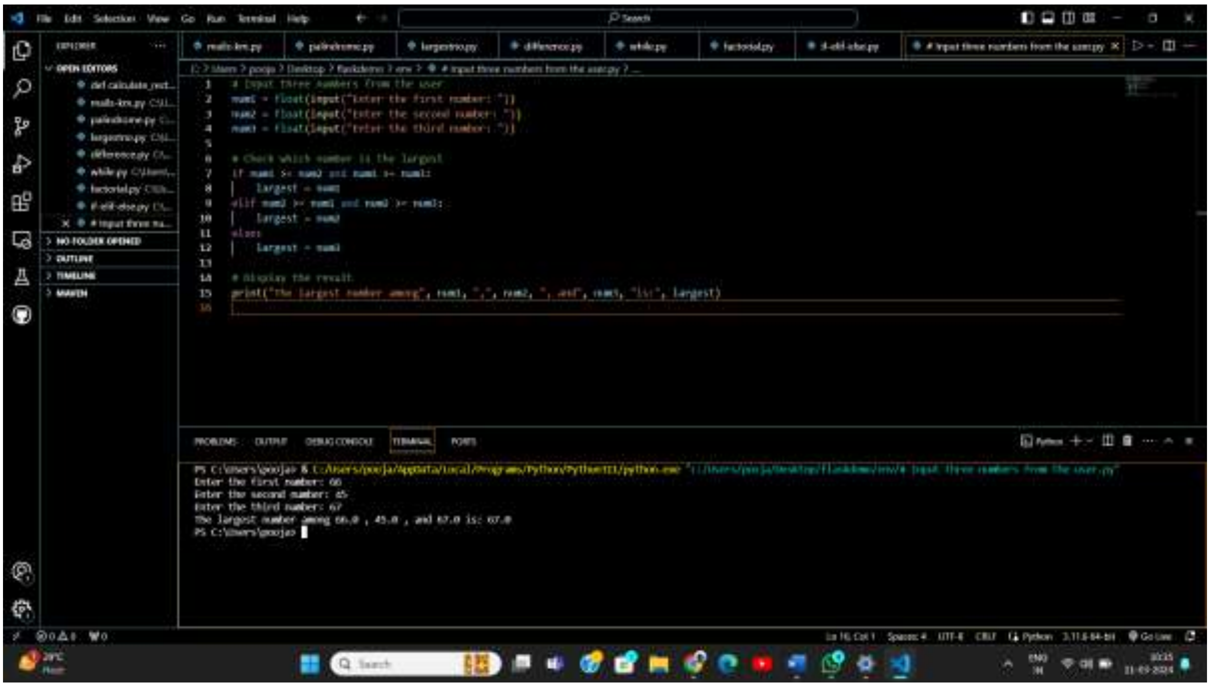
9.write a program to check if a number is positive,negative ,or zero using if-elif-else statements



```
# Input a number from the user
num = float(input("Enter a number: "))

# Check if the number is positive, negative, or zero
if num > 0:
    print("The number is positive.")
elif num < 0:
    print("The number is negative.")
else:
    print("The number is zero.")
```

```
PS C:\Users\pooja> & C:/Users/pooja/AppData/Local/Programs/Python/Python311/python.exe c:/Users/pooja/Desktop/flaskdemo/env/if-elif-else.py
Enter a number: 77
The number is positive.
PS C:\Users\pooja>
```

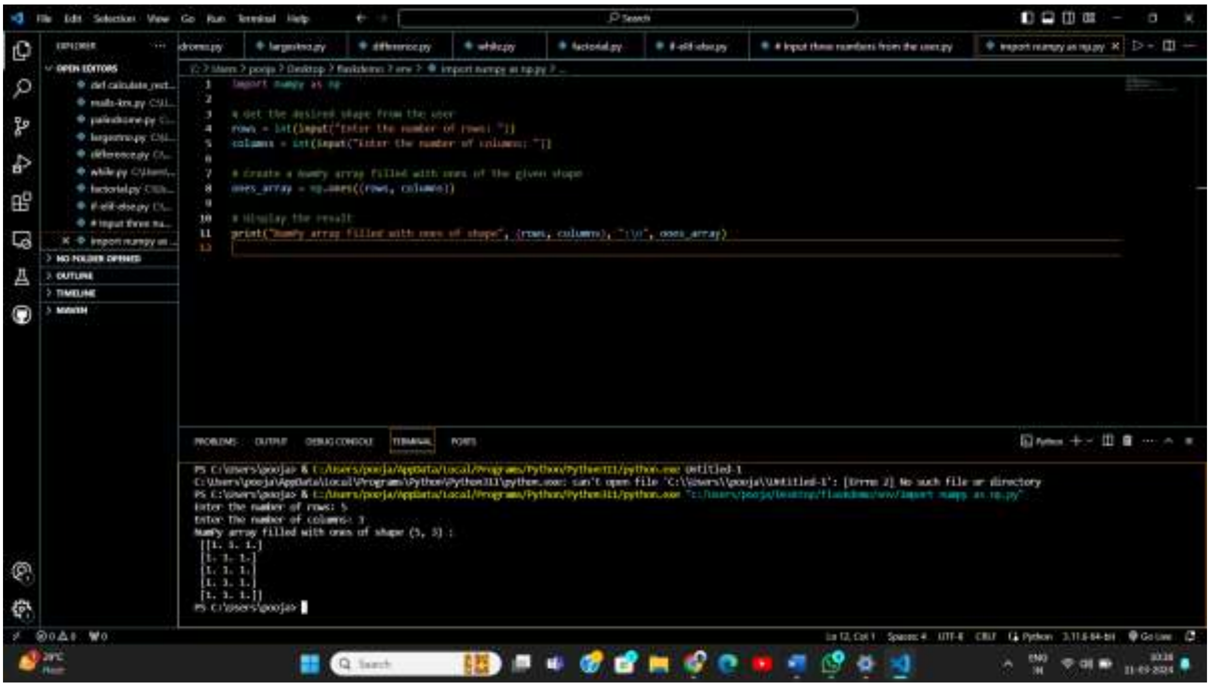
10.write a program to determine the largest among three numbers using conditional statements



```
# Input three numbers from the user
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
num3 = float(input("Enter the third number: "))

# Check which number is the largest
if num1 >= num2 and num1 >= num3:
    largest = num1
elif num2 >= num1 and num2 >= num3:
    largest = num2
else:
    largest = num3

# Display the result
print("The largest number among", num1, ",", num2, ", and", num3, "is:", largest)
```

```
PS C:\Users\pooja> & C:/Users/pooja/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/pooja/Desktop/flaskdemo/env/# input three numbers from the user.py"
Enter the first number: 66
Enter the second number: 45
Enter the third number: 67
The largest number among 66.0 , 45.0 , and 67.0 is: 67.0
PS C:\Users\pooja>
```

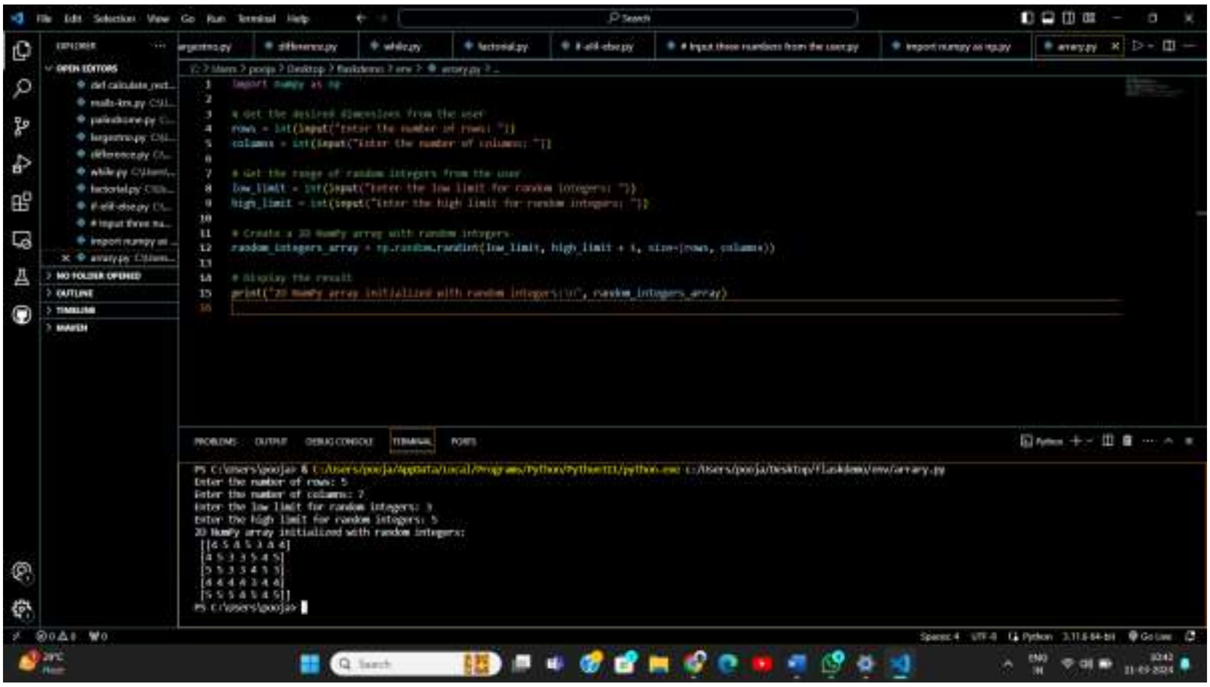
11.write a python program to create a numpy array filled with ones of given shape



```
import numpy as np

# Get the desired shape from the user
rows = int(input("Enter the number of rows: "))
columns = int(input("Enter the number of columns: "))

# Create a NumPy array filled with ones of the given shape
ones_array = np.ones((rows, columns))

# Display the result
print("NumPy array filled with ones of shape", (rows, columns), ":\n", ones_array)
```

```
PS C:\Users\pooja> & C:/Users/pooja/AppData/Local/Programs/Python/Python311/python.exe Untitled-1
C:\Users\pooja\AppData\Local\Programs\Python\Python311\python.exe: can't open file 'C:\\Users\\pooja\\Untitled-1': [Errno 2] No such file or directory
PS C:\Users\pooja> & C:/Users/pooja/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/pooja/Desktop/flaskdemo/env/import numpy as np.py"
Enter the number of rows: 5
Enter the number of columns: 3
NumPy array filled with ones of shape (5, 3) :
 [[1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]]
PS C:\Users\pooja>
```

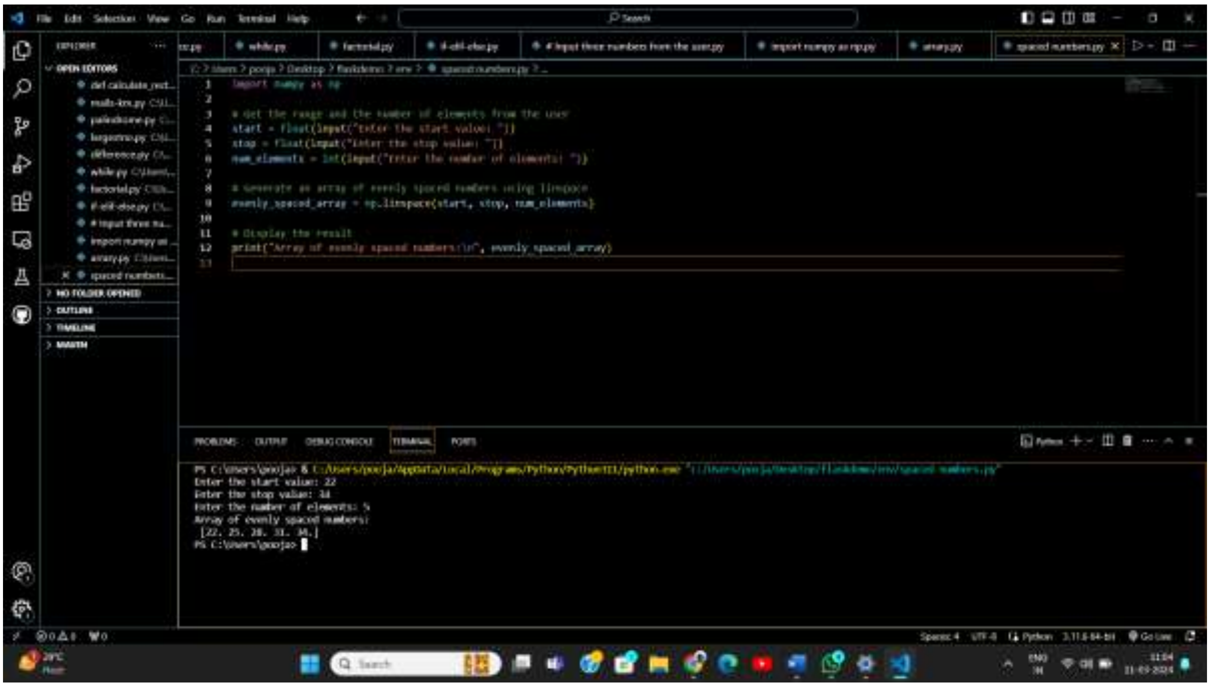
12.write a program to create a 2D numpy array initialised with random integers



```
import numpy as np

# Get the desired dimensions from the user
rows = int(input("Enter the number of rows: "))
columns = int(input("Enter the number of columns: "))

# Get the range of random integers from the user
low_limit = int(input("Enter the low limit for random integers: "))
high_limit = int(input("Enter the high limit for random integers: "))

# Create a 2D NumPy array with random integers
random_integers_array = np.random.randint(low_limit, high_limit + 1, size=(rows, columns))

# Display the result
print("2D NumPy array initialized with random integers:\n", random_integers_array)
```

```
PS C:\Users\pooja> & C:/Users/pooja/AppData/Local/Programs/Python/Python311/python.exe c:/Users/pooja/Desktop/flaskdemo/env/array.py
Enter the number of rows: 5
Enter the number of columns: 7
Enter the low limit for random integers: 3
Enter the high limit for random integers: 5
2D NumPy array initialized with random integers:
 [[4 5 4 5 3 4 4]
 [4 5 3 3 5 4 5]
 [5 5 3 3 4 5 3]
 [4 4 4 4 3 4 4]
 [5 5 5 4 5 4 5]]
PS C:\Users\pooja>
```

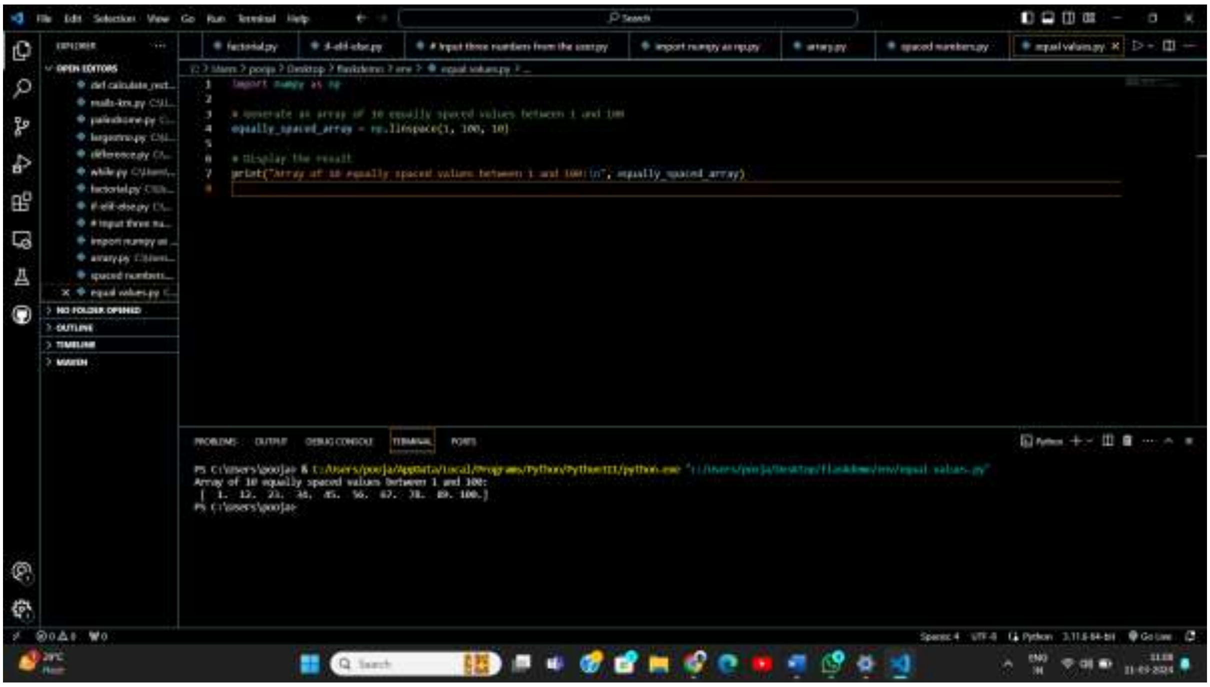
13.write a python program to generate an array of evenly spaced numbers over a specified range using linespace



```python
import numpy as np

# Get the range and the number of elements from the user
start = float(input("Enter the start value: "))
stop = float(input("Enter the stop value: "))
num_elements = int(input("Enter the number of elements: "))

# Generate an array of evenly spaced numbers using linspace
evenly_spaced_array = np.linspace(start, stop, num_elements)

# Display the result
print("Array of evenly spaced numbers:\n", evenly_spaced_array)
```

```
PS C:\Users\pooja> & C:/Users/pooja/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/pooja/Desktop/flaskdemo/env/spaced numbers.py"
Enter the start value: 22
Enter the stop value: 34
Enter the number of elements: 5
Array of evenly spaced numbers:
 [22. 25. 28. 31. 34.]
PS C:\Users\pooja>
```

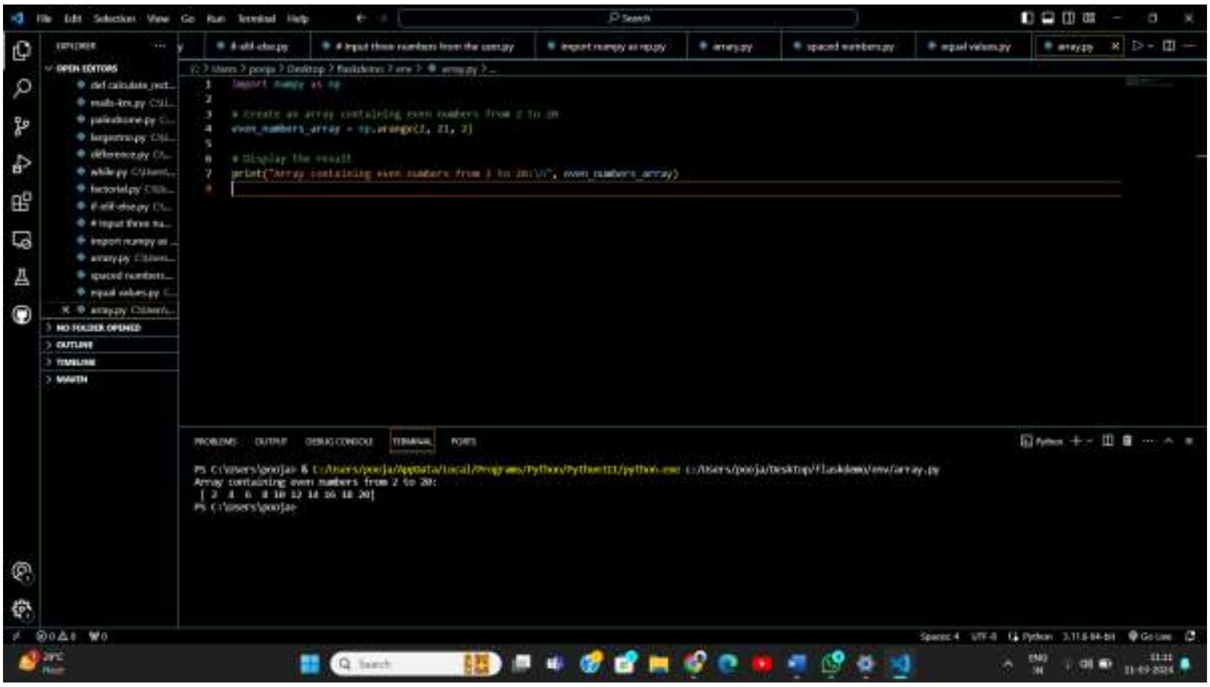
14.write a program to generate an array of 10 equally spaced values between 1 and 100 using linspace



```python
import numpy as np

# Generate an array of 10 equally spaced values between 1 and 100
equally_spaced_array = np.linspace(1, 100, 10)

# Display the result
print("Array of 10 equally spaced values between 1 and 100:\n", equally_spaced_array)
```

```
PS C:\Users\pooja> & C:/Users/pooja/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/pooja/Desktop/flaskdemo/env/equal values.py"
Array of 10 equally spaced values between 1 and 100:
 [  1.  12.  23.  34.  45.  56.  67.  78.  89. 100.]
PS C:\Users\pooja>
```

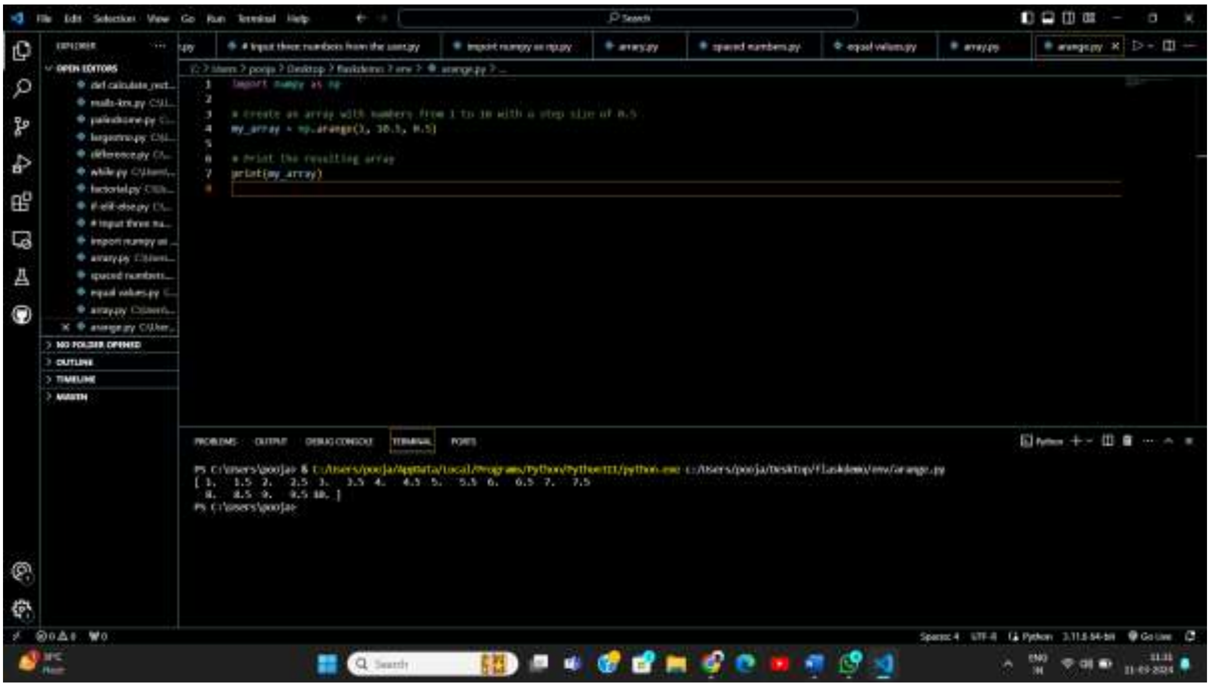
15.write a python program to create an array containing even numbers from 2 to 20 using arrange



```python
import numpy as np

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

# Display the result
print("Array containing even numbers from 2 to 20:\n", even_numbers_array)
```

```
PS C:\users\pooja> & C:/Users/pooja/AppData/Local/Programs/Python/Python311/python.exe c:/Users/pooja/Desktop/flaskdemo/env/array.py
Array containing even numbers from 2 to 20:
 [ 2  4  6  8 10 12 14 16 18 20]
PS C:\users\pooja>
```

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



```python
import numpy as np

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

# Print the resulting array
print(my_array)
```

```
PS C:\users\pooja> & C:/Users/pooja/AppData/Local/Programs/Python/Python311/python.exe c:/Users/pooja/Desktop/flaskdemo/env/arange.py
[ 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. ]
PS C:\users\pooja>
```