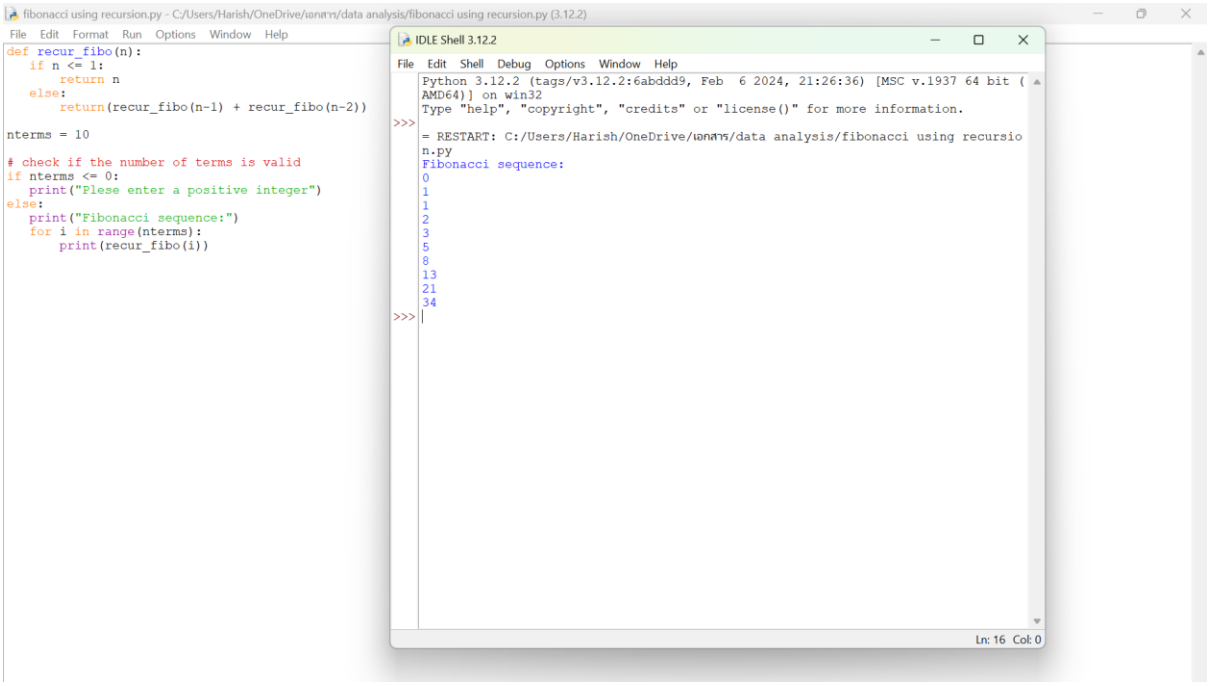
1)Fibonacci Series using recursion.



2) check the given no is Armstrong or not using recursive function.

File Edit Format Run Options Window Help

```python
def order(n):
    count = 0
    while n != 0:
        count += 1
        n //= 10
    return count

def is_armstrong(n, order_val):
    if n == 0:
        return 0
    else:
        return ((n % 10) ** order_val + is_armstrong(n // 10, order_val))

num = int(input("Enter a number: "))
order_val = order(num)

if num == is_armstrong(num, order_val):
    print(num, "is an Armstrong number.")
else:
    print(num, "is not an Armstrong number.")
```

IDLE Shell 3.12.2

File Edit Shell Debug Options Window Help

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:/DAA/LAB PROGRAMS/ARMSTRONG NUMBER.py
Enter a number: 153
153 is an Armstrong number.
>>>
```

3) GCD of two numbers using recursive factorization

File Edit Format Run Options Window Help

```python
def gcd_recursive(a, b):
    if b == 0:
        return a
    else:
        return gcd_recursive(b, a % b)

num1 = 48
num2 = 18

result = gcd_recursive(num1, num2)
print(f"The GCD of {num1} and {num2} is: {result}")
```

IDLE Shell 3.12.2

File Edit Shell Debug Options Window Help

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:/DAA/LAB PROGRAMS/GCD OF 2 NUM.py
The GCD of 48 and 18 is: 6
>>>
```

4) the largest element of an array.

File Edit Format Run Options Window Help

```python
def find_largest_element(arr):
    if not arr:
        return None
    return max(arr)

# Example array
my_array = [10, 5, 20, 8, 15]

largest_element = find_largest_element(my_array)
print(f"The largest element in the array is: {largest_element}")
```

IDLE Shell 3.12.2

File Edit Shell Debug Options Window Help

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:/DAA/LAB PROGRAMS/LARGEST ELEMENT.py
The largest element in the array is: 20
>>>
```

5) the Factorial of a number using recursion.

File Edit Format Run Options Window Help

```python
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

number = 5
result = factorial(number)
print(f'The factorial of {number} is: {result}')
```

IDLE Shell 3.12.2

File Edit Shell Debug Options Window Help

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:/DAA/LAB PROGRAMS/FACTORIAL.py
The factorial of 5 is: 120
>>>
```

6) copy one string to another using recursion

File Edit Format Run Options Window Help

```python
def copy_string(source, destination, index=0):
    if index == len(source):
        return destination
    destination += source[index]
    return copy_string(source, destination, index + 1)

# Test the function
source_str = "Hello, World!"
destination_str = ""
result = copy_string(source_str, destination_str)
print(result)
```

IDLE Shell 3.12.2

File Edit Shell Debug Options Window Help

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:/DAA/LAB PROGRAMS/COPY ONE STRING TO ANOTHER.py
Hello, World!
>>>
```

7) the reverse of a string using recursion

File Edit Format Run Options Window Help

```python
def reverse_string(input_str):
    if len(input_str) == 0:
        return input_str
    else:
        return reverse_string(input_str[1:]) + input_str[0]

# Test the function
input_string = "Hello, World!"
reversed_string = reverse_string(input_string)
print("Original String:", input_string)
print("Reversed String:", reversed_string)
```

IDLE Shell 3.12.2

File Edit Shell Debug Options Window Help

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:/DAA/LAB PROGRAMS/REVERSE OF A STRING.py
Original String: Hello, World!
Reversed String: !dlroW ,olleH
>>>
```

8) generate all the prime numbers using recursion

File Edit Format Run Options Window Help

```python
def is_prime(num, divisor=2):
    if num <= 2:
        return num == 2
    if num % divisor == 0:
        return False
    if divisor * divisor > num:
        return True
    return is_prime(num, divisor + 1)

def generate_primes(n, current=2):
    if current <= n:
        if is_prime(current):
            print(current, end=" ")
        generate_primes(n, current + 1)

# Input the range 'n' up to which you want to generate prime numbers
n = 20
print(f"Prime numbers up to {n}:")
generate_primes(n)
```

IDLE Shell 3.12.2

File Edit Shell Debug Options Window Help

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:/DAA/LAB PROGRAMS/prime numbers.py
Prime numbers up to 20:
2 3 5 7 11 13 17 19
>>>
```

9) to check prime number or not using recursion

File Edit Format Run Options Window Help

```python
def is_prime(n, i=2):
    if n <= 2:
        return True if n == 2 else False
    if n % i == 0:
        return False
    if i * i > n:
        return True
    return is_prime(n, i + 1)

# Input number to check for primality
num = 17

if is_prime(num):
    print(f"{num} is a prime number.")
else:
    print(f"{num} is not a prime number.")
```

IDLE Shell 3.12.2

File Edit Shell Debug Options Window Help

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:/DAA/LAB PROGRAMS/PRIME CHECKING.py
17 is a prime number.
>>>
```

10) to check Palindrome or not using recursions

File Edit Format Run Options Window Help

```python
def is_palindrome(s):
    s = s.lower().replace(" ", "")  # Convert to lowercase and remove spaces
    if len(s) <= 1:
        return True
    if s[0] != s[-1]:
        return False
    return is_palindrome(s[1:-1])

# Test the function
input_string = "A man a plan a canal Panama"
if is_palindrome(input_string):
    print(f"{input_string} is a palindrome.")
else:
    print(f"{input_string} is not a palindrome.")
```

IDLE Shell 3.12.2

File Edit Shell Debug Options Window Help

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:/DAA/LAB PROGRAMS/PALINDROME.py
A man a plan a canal Panama is a palindrome.
>>>
```