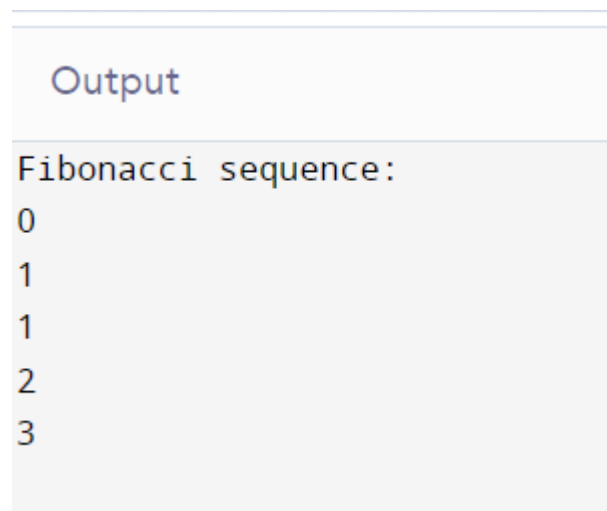


1. Write a program to Print Fibonacci Series using recursion.

```
def fibonacci(n):  
    if n <= 1:  
        return n  
    else:  
        return fibonacci(n-1) + fibonacci(n-2)  
  
nterms = 5  
  
if nterms <= 0:  
    print("Please enter a positive integer")  
else:  
    print("Fibonacci sequence:")  
    for i in range(nterms):  
        print(fibonacci(i))
```

OUTPUT:

A screenshot of a code editor window. The title bar at the top says "Output". The main area of the editor displays the output of the program: "Fibonacci sequence:" followed by the numbers 0, 1, 1, 2, and 3 on separate lines.

```
Output  
Fibonacci sequence:  
0  
1  
1  
2  
3
```

2. Write a program to check the given no is Armstrong or not using recursive function.

```

def order(n):
    return len(str(n))

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

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

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

```

Output

```

Enter a number: 153
153 is an Armstrong number.

=== Code Execution Successful ===

```

3. Write a program to find the GCD of two numbers using recursive factorization.

```

def gcd(a, b):
    if b == 0:
        return a
    else:
        return gcd(b, a % b)

```

```
num1 = 48
```

```
num2 = 18
```

```
result = gcd(num1, num2)
```

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

Output

```
The GCD of 48 and 18 is: 6
```

```
=== Code Execution Successful ===
```

4. Write a program to get the largest element of an array.

```
def find_largest_element(arr):
```

```
    max_element = arr[0]
```

```
    for i in range(1, len(arr)):
```

```
        if arr[i] > max_element:
```

```
            max_element = arr[i]
```

```
    return max_element
```

```
array = [10, 30, 20, 50, 40]
```

```
largest_element = find_largest_element(array)
```

```
print("The largest element in the array is:", largest_element)
```

Output

```
The largest element in the array is: 50
```

```
=== Code Execution Successful ===
```

5. Write a program to find the Factorial of a number using recursion.

```
def factorial(n):  
    if n == 0:  
        return 1  
    else:  
        return n * factorial(n-1)  
  
num = 5  
print("Factorial of", num, "is", factorial(num))
```

Output

```
Factorial of 5 is 120
```

```
=== Code Execution Successful ===
```

6. Write a program for to copy one string to another using recursion

```
def copy_and_print(s, index=0):  
    if index == len(s):  
        return  
    print(s[index], end="")  
    copy_and_print(s, index + 1)  
input_string = "Hello, World!"  
print("Original String:", input_string)  
print("Copied String: ", end="")  
copy_and_print(input_string)  
print()
```

Output

```
Original String: Hello, World!  
Copied String: Hello, World!
```

```
=== Code Execution Successful ===
```

7. Write a program to print the reverse of a string using recursion

```
def reverse_string(s):  
    if len(s) == 0:  
        return s  
    else:  
        return reverse_string(s[1:]) + s[0]  
  
input_string = "Hello, World!"  
reversed_string = reverse_string(input_string)  
print("Original String:", input_string)  
print("Reversed String:", reversed_string)
```

Output

```
Original String: Hello, World!  
Reversed String: !dlroW ,olleH  
  
=== Code Execution Successful ===
```

8. Write a program to generate all the prime numbers using recursion

```
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)
```

```
def generate_primes(n):
    if n > 1:
        generate_primes(n - 1)
    if is_prime(n):
        print(n)
generate_primes(10)
```

Output

```
2
3
5
7

=== Code Execution Successful ===
```

9. Write a program to check a number is a prime number or not using recursion.

```
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)

num = 29
if is_prime(num):
    print(f"{num} is a prime number")
else:
    print(f"{num} is not a prime number")
```

Output

29 is a prime number

=== Code Execution Successful ===

10. Write a program for to check whether a given String is Palindrome or not using recursion

```
def is_palindrome(s):  
    if len(s) <= 1:  
        return True  
    if s[0] == s[-1]:  
        return is_palindrome(s[1:-1])  
    else:  
        return False  
  
input_string = "madam"  
if is_palindrome(input_string):  
    print(f"{input_string} is a palindrome.")  
else:  
    print(f"{input_string} is not a palindrome.")
```

Output

madam is a palindrome.

=== Code Execution Successful ===