Python Programs on Recursion

1. Factorial of a number

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

print(factorial(5))  # Output: 120
```

2. Sum of natural numbers

```
def sum_n(n):
    if n == 0:
        return 0
    return n + sum_n(n - 1)

print(sum_n(10))  # Output: 55
```

3. Fibonacci sequence

```
def fibonacci(n):
    if n <= 1:
        return n
    return fibonacci(n-1) + fibonacci(n-2)

print([fibonacci(i) for i in range(10)])
```

4. Reverse a string

```
def reverse_string(s):
    if s == "":
        return s
    return reverse_string(s[1:]) + s[0]

print(reverse_string("hello"))  # Output: "olleh"
```

5. Count digits in a number

```
def count_digits(n):
    if n == 0:
        return 0
    return 1 + count_digits(n // 10)

print(count_digits(12345))  # Output: 5
```

6. Sum of digits of a number

```
def sum_of_digits(n):
    if n == 0:
        return 0
    return n % 10 + sum_of_digits(n // 10)

print(sum_of_digits(1234))  # Output: 10
```

7. Check if a string is a palindrome

```
def is_palindrome(s):
    if len(s) <= 1:
        return True
    if s[0] != s[-1]:
        return False
    return is_palindrome(s[1:-1])

print(is_palindrome("madam"))  # Output: True
```

8. Print numbers from 1 to N

```
def print_1_to_n(n):
    if n == 0:
        return
    print_1_to_n(n - 1)
    print(n)

print_1_to_n(5)
```

9. Print numbers from N to 1

```
def print_n_to_1(n):
    if n == 0:
        return
    print(n)
    print_n_to_1(n - 1)

print_n_to_1(5)
```

10. Power of a number (x^n)

```
def power(x, n):
    if n == 0:
        return 1
    return x * power(x, n - 1)

print(power(2, 4))  # Output: 16
```

11. Find GCD using recursion

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

print(gcd(36, 60))  # Output: 12
```

12. Binary representation of a number

```
def binary(n):
    if n == 0:
        return ""
    return binary(n // 2) + str(n % 2)

print(binary(10))  # Output: 1010
```

13. Find nth term of arithmetic progression

```
def nth_ap(a, d, n):
    if n == 1:
        return a
    return d + nth_ap(a, d, n - 1)

print(nth_ap(2, 3, 5))  # Output: 14
```

14. Sum of array elements

```
def sum_array(arr, n):
    if n == 0:
        return 0
    return arr[n - 1] + sum_array(arr, n - 1)

print(sum_array([1, 2, 3, 4], 4))  # Output: 10
```

15. Product of two numbers using recursion

```
def multiply(a, b):
    if b == 0:
        return 0
    return a + multiply(a, b - 1)

print(multiply(3, 4))  # Output: 12
```