→ Recursion programs in Python is

1) factorial :-

```
def factorial(n):
    if n==0 or n==1:
        return 1
    else
        return n * factorial(n-1)
print(factorial(5))
```

2)
```
def fibonacci(n):
    if n<=0:
        return "Invalid Input"
    elif n==1:
        return 0
    elif n==2:
        return 1
    return fibonacci(n-1) + fibonacci(n-2)
print(fibonacci(6))
```

3) Sum of natural nos.

```
def sum_natural(n):
    if n==1:
        return 1
    return n + sum_natural(n-1)
print(sum_natural(5))
```

④ Reverse a string

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

print(reverse_string("hello"))
```

```
s = "hello"
reversed_s = s[::-1]
print(reversed_s)
```

⑤ 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("radar"))
print(is_palindrome("hello"))
```

⑥ GCD.

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

print(gcd(48,18))
```

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

(7) Towers of Hanoi

```
def TOH(n, s, a, d):
    if n==1:
        print(f"move disk 1 from {source} to {destination}")
        return
    TOH(n-1, s, d, a)
    print(f"move disk {n} from {source} to {destination}")
    TOH(n-1, auxillary, source, destination)

TOH(3, 'A', 'B', 'C')
```

(8) Binary search using recursion

```
def binary_search(arr, low, high, target):
    if low>high:
        return -1
    mid = (low+high) // 2
    if arr[mid] == target:
        return mid
    elif arr[mid] < target:
        return binary_search(arr, mid+1, high, target)
    else:
        return binary_search(arr, low, mid-1, target)
```

factorial using lambda expression.

```
x = lambda num : 1 if num<=1 else num * x(num-1)
```

using reduce

```
factorial = reduce(lambda x,y : x*y , range(1,n+1))
```

or

```
factorial = lambda number : reduce(lambda x,y : x*y,
                                   range(1, number+1))
```

Fibonacci printing

```
def fibo(i):
  if i<=1;
  return 1
  else;
    return (fibo(i-1) + fibo(i-2))

num=10
  if num<=0:
   print("Please enter a +ive no").
  else:
    print("Fibonacci series:", end=" ")
   for i in range(num):
   print(fibo(i), end=" ").
```