Sum of 1 to N

→ Parameterized [Return x]

→ Functional [Return ✓]

Parameterized

Parameterized

```
func(i, Sum):
    if i<1:
        Print(Sum)
        return
    func(i-1, Sum+i)
```

5 0

func(5, 0)

func(4, 5):
if i<=1:

func(i-1, Sum+i)

func(3, 9)
if i<1:

func(i-1, Sum+i)

func(2, 12)
if i<1

func(i-1, Sum+i)

f(5,0)

↓

f(4,5)

↓

f(3,9)

↓

f(2,12) → f(1,14) → f(0,15) →

func(1, 14)

if i<1:

func(i-1, sum+i)

func(0, 15):

if i<1:

Print(sum)

return

return ? FUNCTIONAL

f(100)

f(1) = 1 → Base condition

f(5) → Return ?

f(1) → Return 1

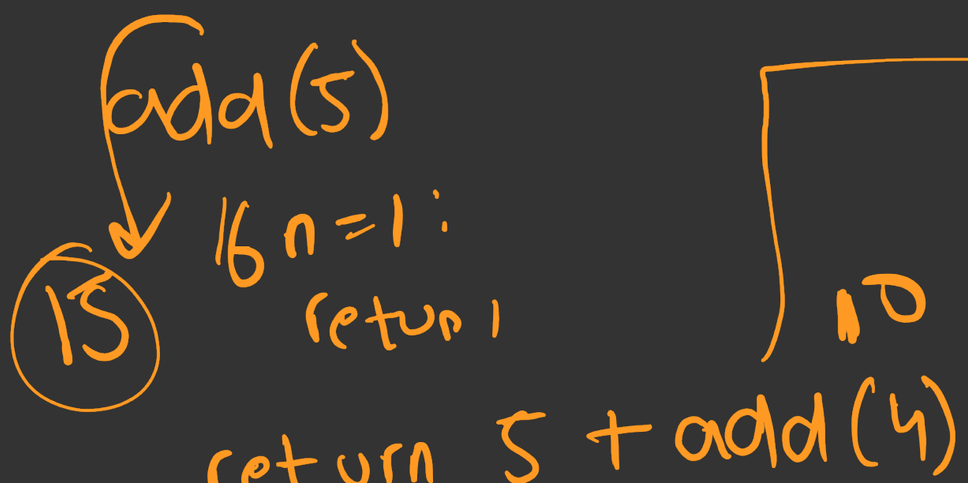
func(5) = 5 + 4 + 3 + 2 + 1

func(5) = 5 + f(4)

func(5) = 5 + 4 + f(3)

func(N) = N + func(N-1)

```python
def addition(n):
    if n == 1:
        return 1
    return n + addition(n - 1)
```



factorial(5) → 5 × 4 × 3 × 2 × 1

fac(5) → 5 × f(4)

factorial(5)

(500)

factorial(1) → 1

factorial(0) → 1

factorial(0) → 1

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

b = 5

p = 3

$5^{\wedge}3 = 125$

$5^{\wedge}3$

$5^{\wedge}4 = 5 \times 5 \times 5 \times 5$

$5^{\wedge}4 = 5 \times 5^{\wedge}3$

func(5,0)

e == 0

return 1

b e

func(5,1)

return b

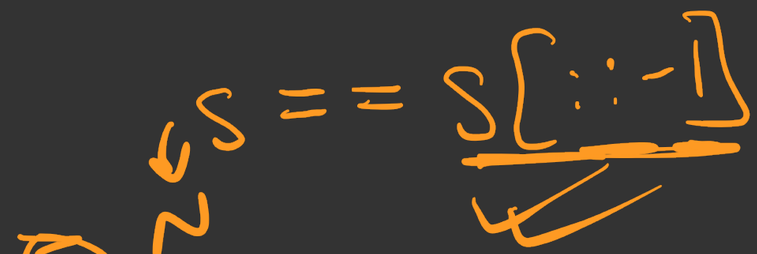
func(5,4) = 5 × 5 × 5 × 5

func(5,4) = 5 × func(5,3)

func(b,e) = b × func(b,e-1)

Using recursion

String S = aeroorea



a[i] != a[L-i-1]

i += 1

L-i-1
8-1-1
6

Fibonacci

7th term

↓

fib(7) → 13

0 ← f(0) f(1)

0 1 1 2 3 5 8 13 21 34 . . . . .

f(2) f(3) f(4) f(5) f(6) f(7) & &

$f(n) = f(n-1) + f(n-2)$

$f(5) = f(4) + f(3)$

f(4) =

f(3) =

(5)

3 2

$f(5) = f(4) + f(3)$

2 1

$f(3) + f(2)$

1 1 0

$f(2) + f(1)$ 1 $f(1) + f(0)$

1 0

$f(1) + f(0)$

1 0

$f(2) + f(1)$

1 1

$f(2) + f(1)$

1 0

$f(1) + f(0)$

= 5