Recursive Functions

```
def func_name(params):
  # code
  pass
```

```
def func_new(a):
  print("Value of a : -->",a)
  if a<=0:
    return a
  else:
    return func_new(a-1)

ret_val = func_new(5)
print("return val: -->", ret_val)
```

```
Value of a : --> 5
Value of a : --> 4
Value of a : --> 3
Value of a : --> 2
Value of a : --> 1
Value of a : --> 0
return val: --> 0
```

```
def factorial(n):
  print("value of n: -->",n)
  if n==0:
    return 1
  return n*factorial(n-1)

ret_val = factorial(5)
print("ret val: -->", ret_val)
```

```
value of n: --> 5
value of n: --> 4
value of n: --> 3
value of n: --> 2
value of n: --> 1
value of n: --> 0
ret val: --> 120
```

```
def sum_n_numbers(n):
  print("n val : -->",n)
  if n==0:
    return 0
  return n+sum_n_numbers(n-1)

ret_val = sum_n_numbers(5)
print(ret_val)
```

```
n val : --> 5
n val : --> 4
n val : --> 3
n val : --> 2
n val : --> 1
n val : --> 0
15
```

Start coding or generate with AI.

```
def sum_n_numbers(n):
  if n==0:
    return 0
  print("n val : -->",n)
  return n+sum_n_numbers(n-1)

ret_val = sum_n_numbers(5)
print(ret_val)
```

```
n val : --> 5
n val : --> 4
n val : --> 3
n val : --> 2
n val : --> 1
15
```

```
def sum_n_numbers(n):
  if n==0:
    return 0
  print("n val : -->",n)
  return n+sum_n_numbers(n-1)

ret_val = sum_n_numbers(5)
print(ret_val)
```

```
n val : --> 5
n val : --> 4
n val : --> 3
n val : --> 2
n val : --> 1
15
```

```
def head_recursion(n):
  if n>0:
    head_recursion(n-1) # head_recursion(1-1)
    print("n val: -->", n) # print(1)

# head_recursion(5)--->head_recursion(1)

def tail_recursion(n):
  if n>0:
    print("n val: -->", n)
    tail_recursion(n-1)



head_recursion(5)
```

```
None
n val: --> 1
None
n val: --> 2
None
n val: --> 3
None
n val: --> 4
None
n val: --> 5
```

```
tail_recursion(5)
```

```
n val: --> 5
n val: --> 4
n val: --> 3
n val: --> 2
n val: --> 1
```

```
def normal_func(n):
  if n>0:
    print(1+5)

normal_func(0)


def finite_recursion(n):
  print("val of n: -->", n)
  if n==0:
    return 0
  return finite_recursion(n-2) # direct recursion


finite_recursion(6)
```

```
val of n: --> 6
val of n: --> 4
val of n: --> 2
val of n: --> 0
0
```

```
finite_recursion(7)
```

```
val of n: --> -1757
val of n: --> -1759
val of n: --> -1761
val of n: --> -1763
val of n: --> -1765
val of n: --> -1767
val of n: --> -1769
val of n: --> -1771
val of n: --> -1773
val of n: --> -1775
val of n: --> -1777
val of n: --> -1779
val of n: --> -1781
val of n: --> -1783
val of n: --> -1785
val of n: --> -1787
val of n: --> -1789
val of n: --> -1791
val of n: --> -1793
val of n: --> -1795
val of n: --> -1797
val of n: --> -1799
val of n: --> -1801
val of n: --> -1803
val of n: --> -1805
val of n: --> -1807
val of n: --> -1809
val of n: --> -1811
val of n: --> -1813
val of n: --> -1815
val of n: --> -1817
val of n: --> -1819
val of n: --> -1821
val of n: --> -1823
val of n: --> -1825
val of n: --> -1827
val of n: --> -1829
val of n: --> -1831
val of n: --> -1833
val of n: --> -1835
val of n: --> -1837
val of n: --> -1839
val of n: --> -1841
val of n: --> -1843
val of n: --> -1845
val of n: --> -1847
val of n: --> -1849
val of n: --> -1851
val of n: --> -1853
val of n: --> -1855
val of n: --> -1857
val of n: --> -1859
val of n: --> -1861
val of n: --> -1863
val of n: --> -1865
val of n: --> -1867
val of n: --> -1869
val of n: --> -1871
```