Recursion

→ A function that calls itself

→ PMI → Principal of Mathematical induction

→ $1 + 2 + 3 \ldots n \Rightarrow \left(\frac{n \times (n+1)}{2}\right)$

Base Condition :

$n = 1$

$1 = \frac{1 \cancel{(1+1)}}{\cancel{2}}$ ✓

L.H.S = R.H.S

2. Induction Hypothesis :
any $k \in n$

$$\boxed{1 + 2 + 3 + 4 \ldots k = \frac{k \times (k+1)}{2}}$$

3. Proove (Induction Step)
$(k+1)$ ✓

$(k+1)^2$

$$\underbrace{(1+2+3 \ldots +k)}+(k+1) = \frac{(k+1) * (k+1+1)}{2}$$

$$\Rightarrow \frac{(k+1) * (k+2)}{2}$$

$$\frac{k * (k+1)}{2} + k+1$$

R.H.S

$$\left[\frac{k^2+k+2k+2}{2}\right]$$

$$\frac{k(k+1)+2(k+1)}{2}$$

$$\frac{(k+1)(k+2)}{2} = \frac{(k+1)(k+2)}{2}$$

L.H.S = R.H.S

To solve any Question with Recursion

1. Base Condition
2. Recurrence Relationship

→ equation 1

$$fun(n) = \underbrace{1 + 2 + 3 \cdots (n-1)}\, n$$

② $f(n) = \boxed{n + f(n-1)}$

① $F(1) = 1$

```
Def func(n)
    if n==1 :
        return 1
    else:
        return n + func(n-1)
```

$f(4)$

1
$2 + F(1)$ $= 3$
$3 + F(2)$ $= 6$
$4 + f(3)$ $= 10$
$f(4) \rightarrow 10$

$\cancel{F(2) = 3}$

$\boxed{\text{Ex-2}}$

2 2 2 2

→ $1^2 + 2 + 3 + 4 \ldots n^-$

→ Base Cond → $f(1) = 1$

→ Recurrence Relation

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

```
def func(n):
    if n == 1:
        Return 1
    else:
        return n^2 + f(n-1)
```

$$1 + \frac{1}{2} + \frac{1}{3} + \ldots \frac{1}{n}$$

① Base Condition

$$f(1) = 1$$

② Recurrece R

$$\frac{1}{n} + f(n-1) \leftarrow$$

$$\sim \left(\frac{1}{n-1}\right)$$

→

Ex -3

$f(n, p) \Rightarrow 4, 3$

4^3

$\Rightarrow$ Base condition

$f(n, 1) \Rightarrow n$ ✓

$f(n, 0) \Rightarrow 1$ ✓

$\Rightarrow$ Recurren Relationship

$$\boxed{n \times (F(n, p-1))}$$

$\Rightarrow$

```
def func_pow(n, p):
    if p == 1:
        return n
    else:
        return n * func_pow(n, p-1)
```

$f(2,1)$
$f(2,2)$
$\cancel{f(2,3)}$

② $2 \times (\cancel{f(2,1)}) \Rightarrow 2 \times 2 = 4$

$\Rightarrow 2 \times \cancel{f(2,2)} \Rightarrow 2 \times 4$

$\Rightarrow 8$

⇒ HW ⇒ 1, 2, 3 ✓

$f(4)$ { $1^2 + 2^2 + 3^2 + n \ldots + n^2$ ×

$1 + \frac{1}{2} + \frac{1}{3} \;---\; \frac{1}{n}$ ×

$f(2,3)$ ✓ Pow

[① Base-Condition
Recursion R]

② all subsets ✓

⇒ [1, 2, 3]

[ ]
[1, 2]

② A, B =>
[1, 2
[1, 3

```
def fun(i, j):
    if i == 0:
        return j
    else:
        ret fun(j, i-5)
```

fun(15, 8) ←

f(0, -7)
f(-2, 0)
f(5, -2)
f(3, 5)
f(10, 8-5)
f(8, 15-5)
f(15, 8)

= -7

f(n)
O(1)

5

→ Count = 'hi'
String = abbhicchiccd
[2]

def Counthi(string, count=0)
1) if len(string)<2:
return Count

2) else:

R·R→

if string == 'hi': }
return Count + 1 }

'hi' (
↔

C... hiabc → P(abc, c+1)

$s\ (x + y) = a$ hiabc $\Rightarrow f(abc, c)$

$f = 1$ ahiabc $\Rightarrow f($hiabc$, c)$

aahiabc $\Rightarrow f$

$f = 1$ (ahiab, c

2

(2)

first 2 letters are Hi

→ reduce string by 2

→ increment count

X f(2) not 'Hi'

→ reduce string by 1

→ no → c.

hishi

→ count +1

f (ishi) S[1: X

f(shi) S[2:

aaa ⇒ 2̶

aaa ⇒ 1, 'aa' jumpsize = :2

'aaa' ⇒ 2, 'aa' jumpsize = :1

hihisc

f(hihisc) ✗ can ✗ jum

f(hisc)

⇒ 10, (10−2, 10−4)

⇒ 10, (8, 6, −−−

⇒ 10 + (f(8)

$\Rightarrow 10 + (f(10-2) +$

$n + (n-2) \ldots$

$f(8) = 8 + f(6)$

$10 + f(8) \Rightarrow 8 + f(8-2)$

$f(9), f(7), f(5), f(3)$ $f(1), f(-1)$

$\Rightarrow$ if $n < 1 =$
return 0

if $n == 0$
return 0