* Armstrong Number: → Retrieve single digits from a no.

321 → (1,2,3) → Counting the no. of digits.

153 → 3 digits → $1^3+5^3+3^3$

$n = n/10$ $15/10 = 1/10 = 0$

0

$1^3+3^3+5^3 = (153)$

cd(n) (3)

153

$n \% 10 = 3$

$Sum = 0+3^3 = 9$

$9+5^3$ $9+125+1^3$

Factorial of any Positive Number : →

$5! = 5\times4\times3\times2\times1 = 120$

$n=5 \longrightarrow 1\times2\times3\times4\times5$

$0! = 1$

$1! = 1$

{ Sum of numbers 1 to N }

① For Loop

```
Sum = 0
for(i = 1; i<=n; i++)
  Sum += i;
```

② While Loop

```
Sum = 0
While (N-- > 0){
  sum += N;
}
```

③ Formula

$\left(\frac{n \times n+1}{2}\right)$

Sum : → N = 5

5 + 4 + 3 + 2 + 1

n → n-1 or n-2 (n → prev) { previous value }

* Recursion: A function calling itself is called recursion until a base case is reached.

(n == 0 return 0)

Sum(10) →

Recursion Tree

10 + Sum(10-1) (9)

10 + 9 + Sum(8)

10 + 9 + 8 + S(7)

10 + 9 + 8 + ... + S(0)

n == 0 (no) (stop)

Base case

Fibonacci Series: →

0 1 2 3 4 5 6 7 8

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

f(0) = 0

f(1) = 1

f(2) = f(1)+f(0) = 0+1 = 1

f(3) = f(2)+f(1) = 1+1 = 2

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

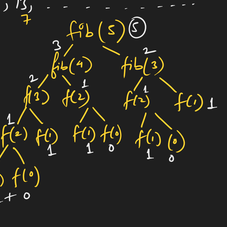
Recursion Tree for Fibonacci Sequence : →

0, 1, 1, 2, 3, 5, 8, 13, ...

0 1 2 3 4 5 6 7

f(0) = 0

f(1) = 1



Just Variables : → int p2 = 0, p1 = 1;

(from 2 to n)

Unknown

```
for(int i = 2; i<=n; i++){
  int curr = p2+p1;
  p2 = p1;
  p1 = curr;
}
return p1;
```

p2	p1	curr
0	1	1
1	1	2
1	2	3
2	3	5
3	5	8

→ curr = p2 + p1

→ p2 = p1

→ p1 = curr