Fibonacci Series using Recursion

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

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

```
public static int fibSeries(int n){
        if(n<=1){
          return n;
        }
      return fibSeries(n-1) + fibSeries(n-2);
    }
P s v main(String arr[]){
    for(int i=0; k=10; i++){
      Sop(fibSeries(i));
    }
}
```

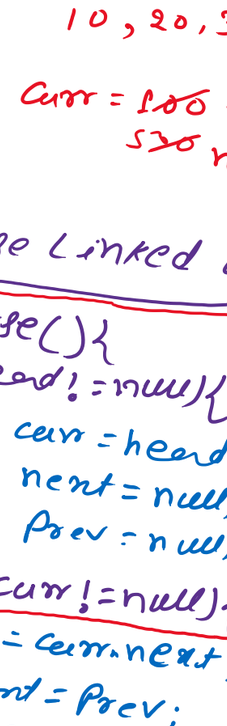
```
fibonacciSeries withoutRecursion(int n){
int i=0;
int j=1.
int k=0;
        =n){
Sop(k);
i=j;
j=k;
k=i+j;
}
}
```

k=0 1 1
j=0 1 0
j=1 0 1

//0,1,1

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

```
public static int fib(int n){
        if(n<=1){
          return n;
        }
      return fib(n-1) + fib(n-2);
    }
Public static void main(String arr[]){
    for(int i=0; i<=10; i++){
     Sop(fib(i));
    }
}
```



i=0 1 2 3 4
n=0, 1 2 3 4

$fib(n-1) + fib(n-2)$

$fib(1) + fib(0)$

$1 + 0 \Rightarrow 1$

$fib(n-1) + fib(n-2)$

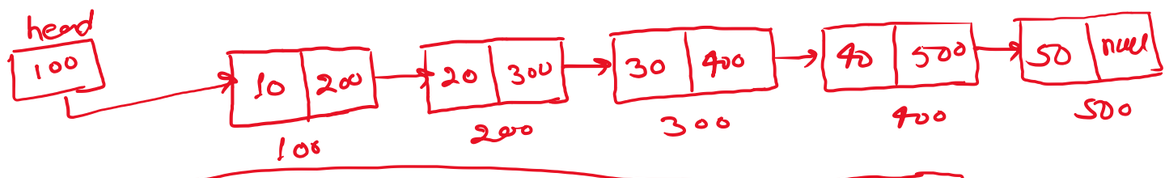
$fib(2) + f(1)$

$1 + 1 = 2$

$fib(n-1) + fib(n-2)$

$f(3)$

Print Singly linked list using Recursion ⇒



10, 20, 30, 40, 50

curr = 100 200 300 400 500 null

```
pubic void printDataUsingRecursion(Node curr){
        if(curr == null){
          return;
        }
        Sop(curr.data);
        curr = curr.next;
        printDataUsingRecursion(curr);
}
```

Reverse Linked List

```
reverse(){
If(head!=null){
Node curr=head;
Node next=null;
Node Prev=null;
while(curr!=null){
    next=curr.next;
    Curr.next=Prev;
    Per=Curr;
    Curr=next;
}
if(Pre!=null){
      head=prev;
}
}
```

1 → 2 → 3 → 4

null ← 1 ← 2 ← 3 ← 4

24-07-2023 Covered topic

① fibonacci series using Recursion.
② Print Linked List using Recursion

25-07-2023 topic

① back tracking in recursion
② program using recursion

Reverse Linked by two way

① We can change only address and data will be same.
② Revere only data and address will be not change