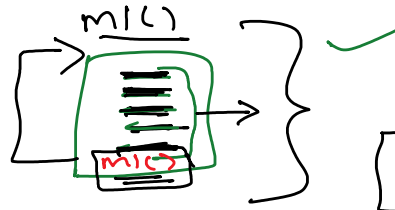
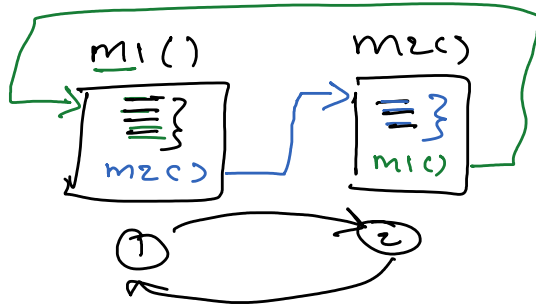


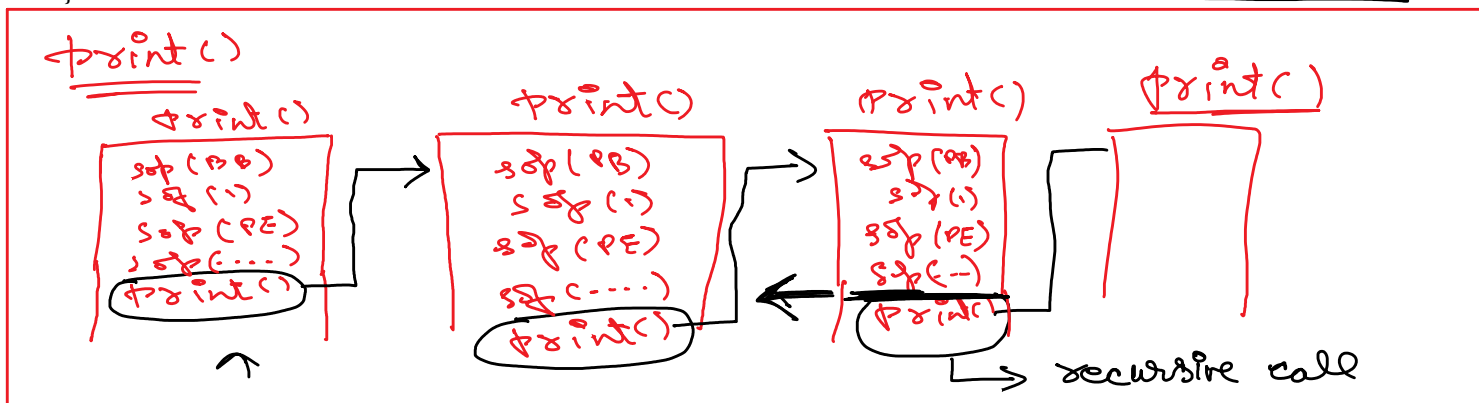
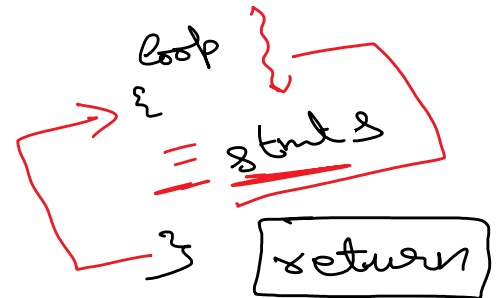
method

m1() ← ① once  
recursive call



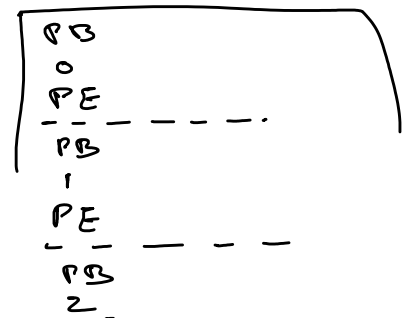
m1()

```
static void print()
{
    System.out.println("Print Begin ");
    System.out.println(1);
    System.out.println("Print end ");
    System.out.println("=====");
    print();
}
```

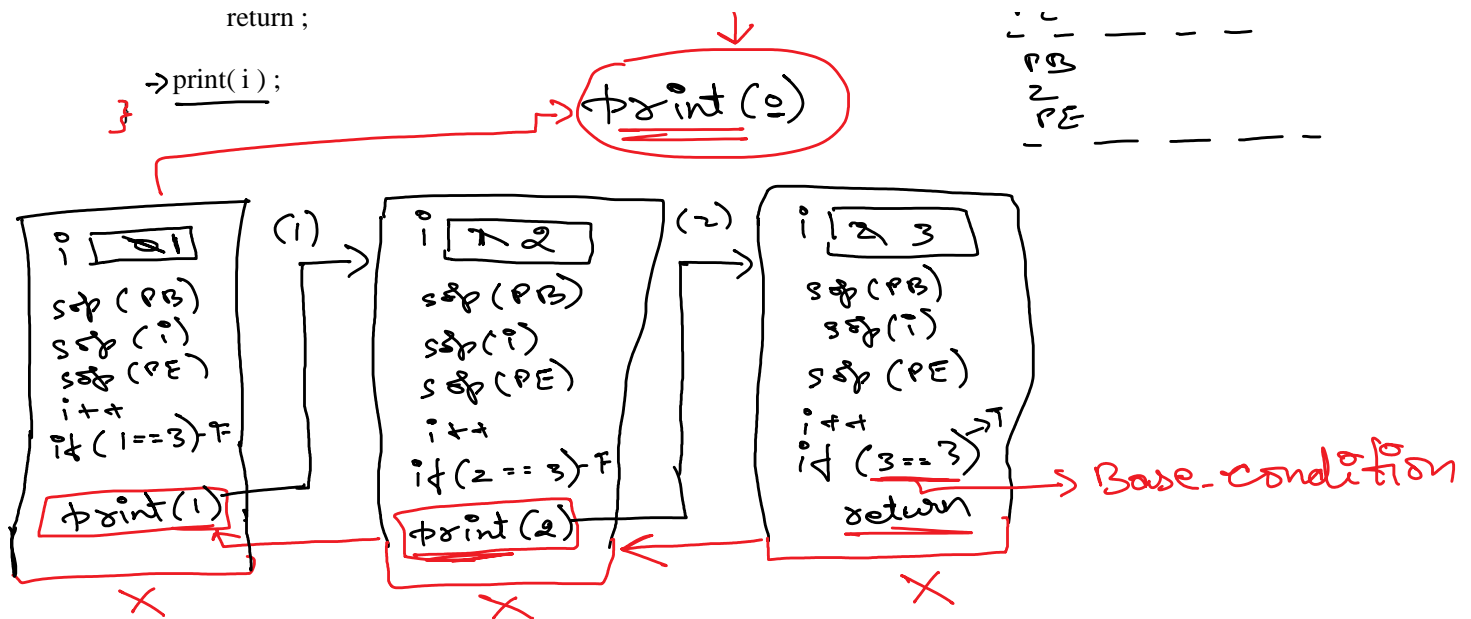


Stack Overflow Error — Stack

```
static void print(int i)
{
    System.out.println("Print Begin ");
    System.out.println(i);
    System.out.println("Print end ");
    System.out.println("=====");
    i++;
    if (i == 3)
        return;
    print(i);
}
```



→ print(i) ;  
print(0)



```

static void printN( int n , int m )
{

```

```

    System.out.println( n++ );
    if ( n == m )
        return ;
    printN( n , m );
    System.out.println( n );
}

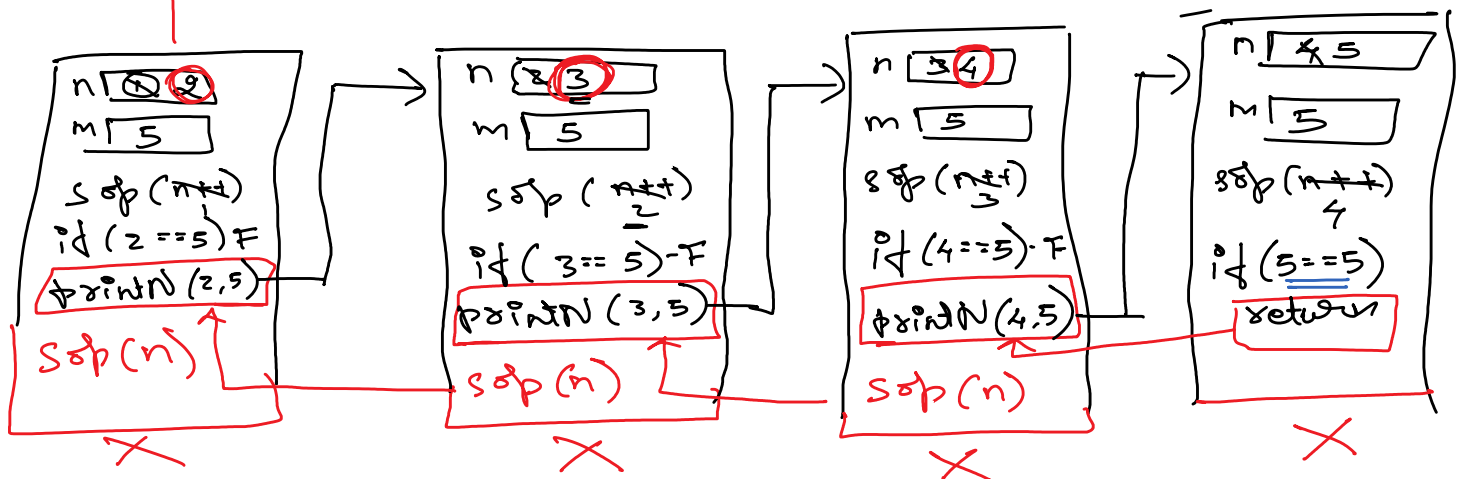
```

Before recursive call

after recursive call

1 2 3 4 4 3 2

printN(1, 5)



Tips To design Recursive solution for Problem Statement.

1.) identify repetitive stmts ..

1.) identifying

2.) identify the Base condition  
(identify when to stop repetition)

Program to find  $n^{\text{th}}$  factorial

$$1! = 1$$

$$2! = 1 \times 2$$
$$2! = 1! \times 2$$

$$3! = 1 \times 2 \times 3$$
$$= 1! \times 2 \times 3$$
$$3! = 2! \times 3$$

$$4! = \frac{1! \times 2}{2! \times 3} \times 4$$

repeatedly  
→ till  $n=1$

$$(n-1)! \times n$$

$$5! = (5-1)! \times 5$$

120

$$(4-1)! \times 4$$

24

$$(3-1)! \times 3$$

6

$$(2-1)! \times 2$$

2

$$1!$$

1

int fact (int n)

{ if (n == 1) return 1;

{  
if (n == 1) return 1;  
return fact(n-1) \* n;  
}