$PI(n)$ | $PI(3) \checkmark \longrightarrow$ $\dfrac{1}{2}$
$\dfrac{3}{n}$

$\begin{array}{c}1\\2\\3\\n\end{array}$
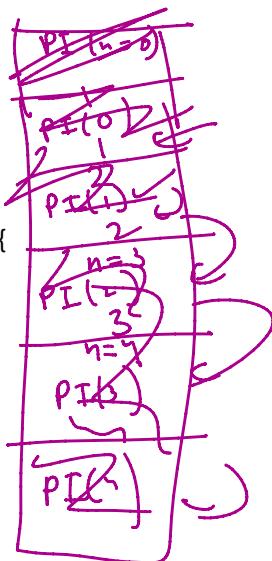
$PI(4)\{$

$PI(3)$
$\}$ $Spo(n)$

```
    PI(4);
    }

    public static void PI(int n) {
//   BP : PI(n)
//   SP : PI(n-1)
     if(n==0) {
        return;
      }
     PI(n - 1);//
     System.out.println(n);
    }
```

$\begin{array}{c}1\\2\\3\\n\end{array}$

$\Big(\quad S, e \quad \Big)$

———

$PID(4)$ | $PID(3)$ | $PID(2)$ | $PID(1)$ | $PID(4)\{$

$\begin{array}{c}4\\3\\2\\1\\ \frac{1}{2}\\3\\n\end{array}$ | $\begin{array}{c}3\\2\\1\\ \frac{1}{3}\end{array}$ | $\begin{array}{c}2\\1\\ \frac{1}{2}\end{array}$ | $\begin{array}{c}1\end{array}$ | $PID(3)$

$BP \rightarrow PID(n)$

$Sypo(n)$

$PID(n-1) \longrightarrow$

n-1
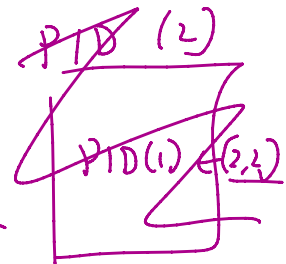n-2
3
2
3
n-1

$Sypo(n)$
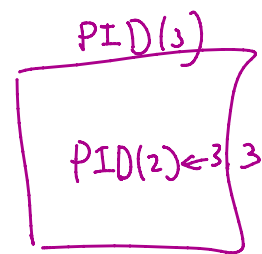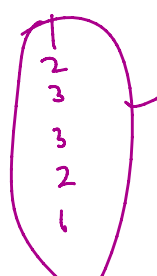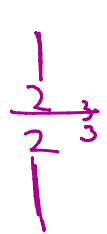
```
public static void PID(int n) {
    if(n==0) {
        return;
    }
    System.out.println(n);
    PID(n-1);
    System.out.println(n);
}
```
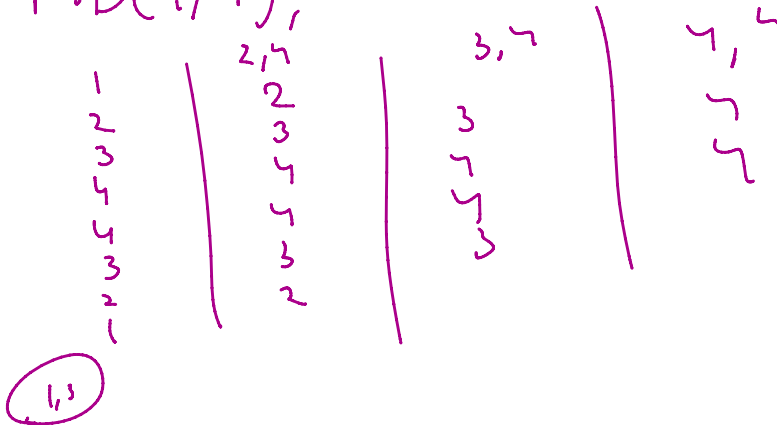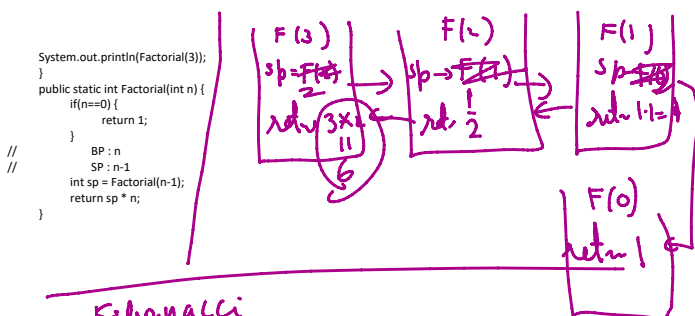
PID
PID(1)
2 3
PID(2)
PID(3)
PID(1)

4
3
2
1

1 2 3 4

PID $PID(4)$

1
2
3
4
4
3
2
1

$PID(3)$

1
2
3
3
2
1

$PID(2)$

1
2
2
1

$PIO(1)$

1
1

$PID(n) \{$

$PID(n-1) \leftarrow$ $\frac{n}{3}$

3

1 2 3
2 3

$PID(3)$

$PID(2) \leftarrow 3$ 3 $\leftarrow$

$PID(2)$

$PID(1) \leftarrow (2,1)$

2
3
3
2
1

$PID(1)$

J

$\frac{1}{2}$ 3
$\frac{2}{2}$ 3

2
3
3
2
1

PID()

PID(1, 4);

```
1          2,4         3,4          4,4
2           2           3            4
3           3           3
4           4           5
4           5           3
3           3
2           2
1
```

(1,5)
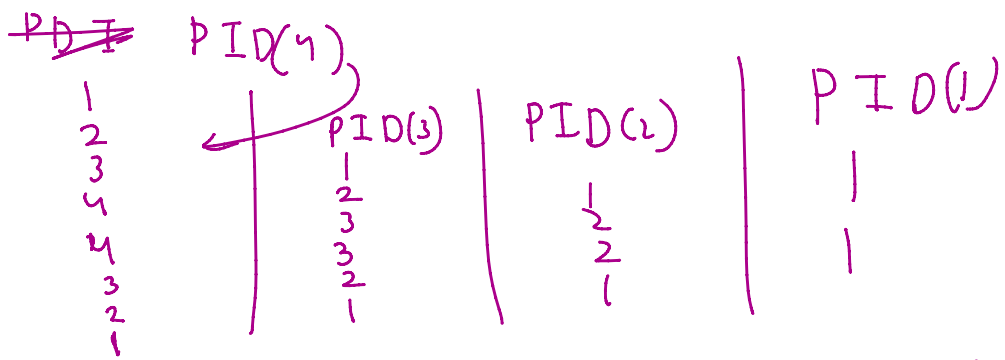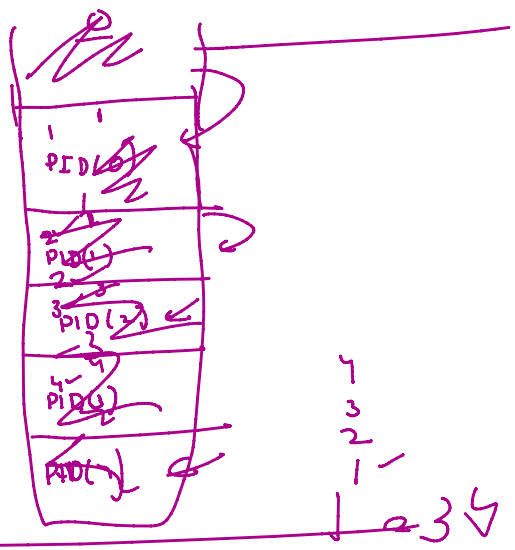
```java
public static void PDI(int s, int e) {
    if(s>e) {
        return;
    }
    System.out.println(s);
    PDI(s + 1, e);
    System.out.println(s);
}
```

---

n!

5! = 1 x 2 x 3 x 4 x 5

4! = 1 x 2 x 3 x 7

```java
System.out.println(Factorial(3));
}
public static int Factorial(int n) {
    if(n==0) {
        return 1;
    }
//      BP : n
//      SP : n-1
    int sp = Factorial(n-1);
    return sp * n;
}
```

F(3)   F(2)   F(1)
sp=F(2)   sp→F(1)   sp=1
  2
rdn 3x   rdn 2   rdn 1·1=1
  11
  6

F(0)
return 1

Fibonacci

0, 1, 1, 2, 3, 5, 8, 13, 21, ?
0 1 2 3 4 5 6 7   8   9

int Fibo(7){

    sb1 = Fibo(6)
    sb2 = Fibo(5)
    rets sb1+sb2;

$sp1 = Fibo(6)$

$sp2 = Fibo(5)$

ret> $sp1 + sp2$;

```
public static int Fibo(int n) {
    if (n <= 1) {
        return n;
    }
    //      BP : F(n)
    //      SP : F(n-1) F(n-2)
    int sp1 = Fibo(n - 1);
    int sp2 = Fibo(n - 2);
    return sp1 + sp2;
}
```

② n=4
$sp1 = F(3) = 2$
$sp2 = F(2) = 1$
ret $1+2 = 3$

③ n=3
$sp1 = F(2) = 1$
$sp2 = F(1) = 1$
ret $1+1 = 2$

F(2)
$sp1 = F(1) = 1$
$sp2 = F(0) = 0$
$= 1 + 0$

⑦ n=2
$sp1 = F(1) = 1$
$sp2 = F(0)$
ret $1+0 = 1$

⑤ n=1
return 1

n=1
return 1

n=0
0

ret 1

set 0

F(3)
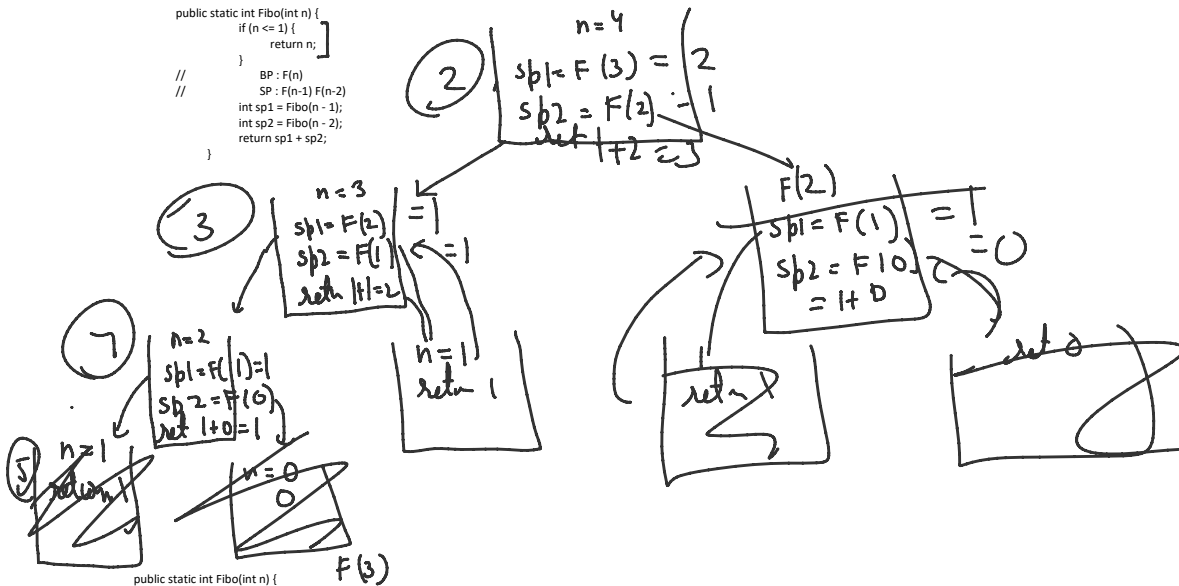
```
public static int Fibo(int n) {
    if (n <= 1) {
        return n;
    }
    int sp1 = Fibo(n - 1);
    int sp2 = Fibo(n - 2);
    return sp1 + sp2;
}
```

m
Fibo(3) = 2

F(3)
F(2)
F(1)
F(1)
F(0)
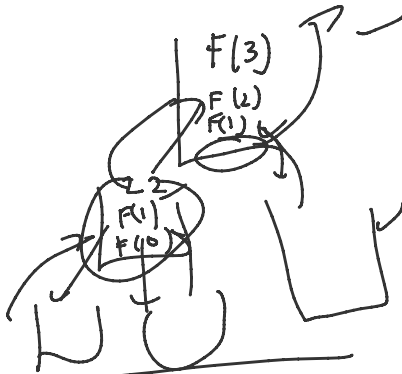
Array rec
Q Print

$\{10, 20, 30, 40, 50\}$
 0   1   2   3   7

$\{20, 30, 70, 50\}$

$\{10, 20, 30, 40\}$

10k

```
public static void print(int[] arr, int idx) {
    if(idx==arr.length) {
        return ;
    }
    System.out.println(arr[idx]);
    print(arr, idx+1);
}
```

10k, 0
100
P(1)

20
P(2)

14=fn

40

2

30

40
2
4=4   P(4)   30
          P(3)

$\{ \overset{0}{10}, \overset{1}{20}, \overset{2}{30}, \overset{3}{40} \}$

→ 40, 30, 20, 10.

Rev (0) →
  40, 30, 20, 10

Rev (1) →
  30, 20, 10   40, 30, 20

$\overset{0}{} \quad \overset{1}{} \quad \overset{2}{} \quad \overset{3}{} \quad \overset{4}{}$

$\{ \overset{0}{5}, \overset{1}{10}, \overset{2}{6}, \overset{3}{11}, \overset{4}{9} \}$

11
all 10
sp = (1)   sp = (2)
11   11

3   2
sp = 9   sp = 11   11
11   11

4
sp = -∞
ret 9
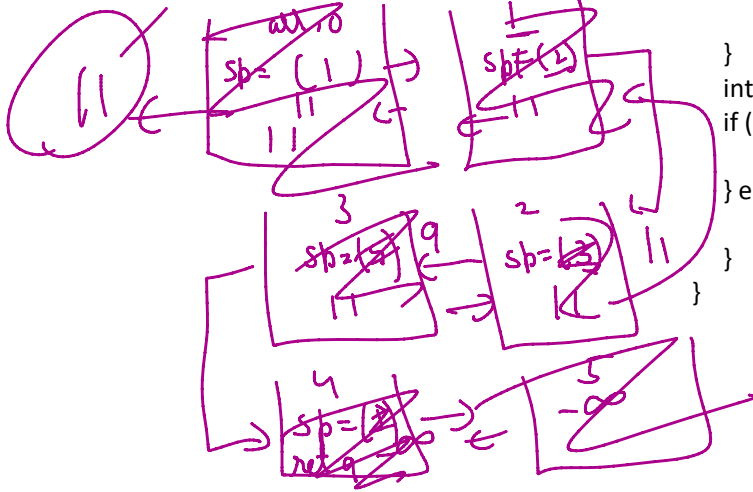
```
public static int Max(int[] arr, int idx) {
    if(idx==arr.length) {
        return Integer.MIN_VALUE;
    }
    int sp = Max(arr, idx + 1);
    if (sp > arr[idx]) {
        return sp;
    } else {
        return arr[idx];
    }
}
```
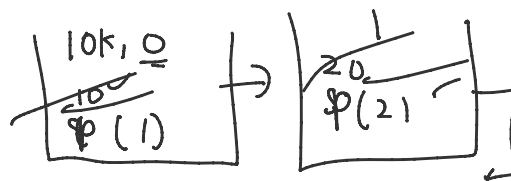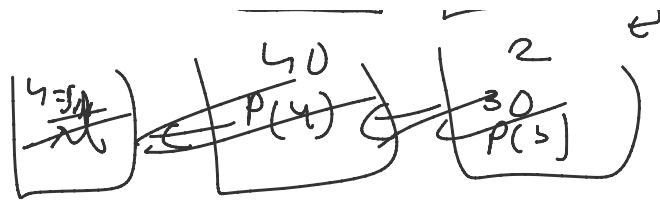
Find.

{

$\{ \overset{0}{10}, \overset{1}{20}, \overset{2}{20}, \overset{3}{30}, \overset{4}{20}, \overset{5}{40} \}$

Q1) First OCC → int   Q3)
                                int[]
Q2) Last OCC → int | 1 | 2 | 4 |

Q2) Lat OCC →int || | 1 | 2 | 4 |