

---

---

---

---

---



## # functions

$$f(x) = x + 5;$$

input = 2

output = 7

$$f(a, b) = a + b;$$

input = 2, 3

output = 5

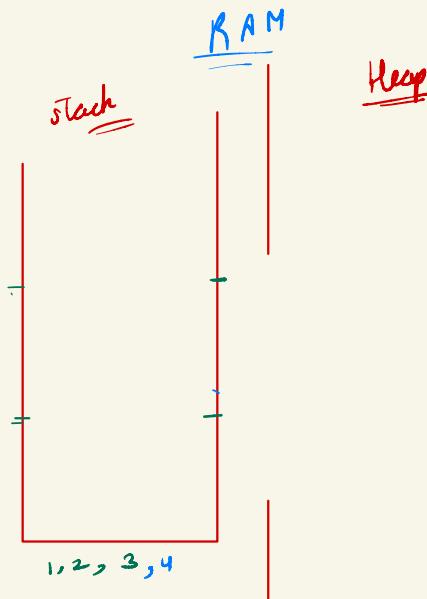
⇒ public static int sum(int a, int b){  
    int sum = a + b;  
    return sum;

}

~~public static~~ action-type void fun-name (parameters) {  
    // logic

3

```
public class Main {  
    public static int sum(int a, int b){  
        int sum = a + b;  
        return sum;  
    }  
  
    public static void main(String[] args) {  
        1 int a=2;  
        2 int b=3;  
  
        3 int ans=sum(a, b);  
        4 System.out.println(ans);  
    }  
}
```



```

public class Main {
    public static void printSomething(){
        System.out.println("Hi!!!");
        printSomething(),
        int a=5;
        a++;
        a--;
        a=11;
    }

    public static int sum(int a, int b){
        printSomething();

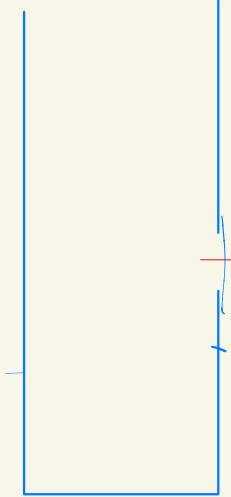
        int sum = a + b;
        return sum;
    }

    Run | Debug
    public static void main(String[] args) {
        int a=2;
        int b=3;

        int ans=sum(a, b);
        System.out.println(ans);
    }
}

```

Hi!! ↘  
5 ↘



## # PNI Principal of Mathematical Induction

sum of first  $n$  natural numbers  $\Leftarrow$

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

$n=0, n=1 \rightarrow$  base case  
 suppose  $\underline{\underline{n=k}} \rightarrow$  smaller problem  
 $\underline{\underline{n=k+1}} \rightarrow$  solving

## # Recursion

a function calling itself

- 1) solve for the smallest problem
- 2) keep a faith that it will work for a smaller problem. (faith)
- 3) solve for rest of the problem. (to reach your expectation)

## # print decreasing

- 1)  $n=1$  is the smallest problem
  - 2)  $n=s$ ,  $(n-1)$
- 

if ( $n==1$ )  
    System.out.println(1);  
    return;

## Faith (smaller value)

$$n-1 \Rightarrow 4$$



Expectation  
If I provide  $n=5$ , it will

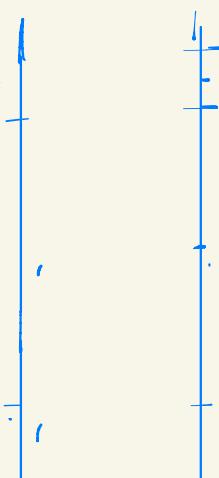


dry run is very important

10 lines  $\rightarrow$  2 pages of notebook

```
public static void pd(int n){
    if(n==0){
        return;
    }

    System.out.println(n);
    pd(n-1);
}
```



ans print increasing recursively  
 $n=5$

1) you won't be able to learn recursion if you don't dry run.

Faith  
It will print numbers  
for (n-1);  
 $n=4 \Rightarrow \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$

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

expectation  
1) my function should print increasing numbers.  
2) for  $n=5$ , it should print  
1  
2  
3  
4  
5

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

break till  
 $\rightarrow [10:15]$

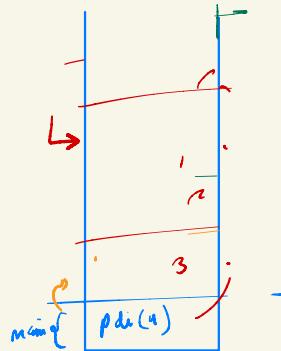
# print decreasing numbers

faith

$n-1$

$n=3$   
3  
2  
1  
1  
2  
3  
4  
3  
2  
1  
1  
2  
3

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



expectation  
It will print numbers in decreasing, then increasing.

$n=4$

4  
3  
2  
1  
1  
2  
3  
4

## Find factorial of a number

Faith

it will give me  
 $\text{fac}(n-1)$ ;

$$4 \times 3 \times 2 \times 1$$

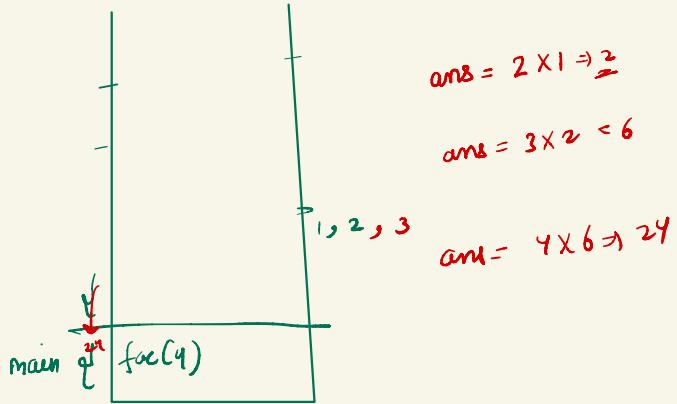
```
public static int fac(int n) {
    1 if (n == 1) return 1
    2 int sAns = fac(n-1);
    3 int ans = n * sAns;
    4 return ans;
}
```

expectation

My function will return  
factorial of  $n$

$$\underline{\underline{n=s}}$$

$$\frac{5 \times 4 \times 3 \times 2 \times 1}{\cancel{1}}.$$



$$\text{ans} = 2 \times 1 \Rightarrow 2$$

$$\text{ans} = 3 \times 2 \Rightarrow 6$$

$$\text{ans} = 4 \times 6 \Rightarrow 24$$

H.W sum of  $n$  natural numbers using recursion.

$$n=5$$

$$1+2+3+4+5 \Rightarrow 15$$