Recursion.

↓

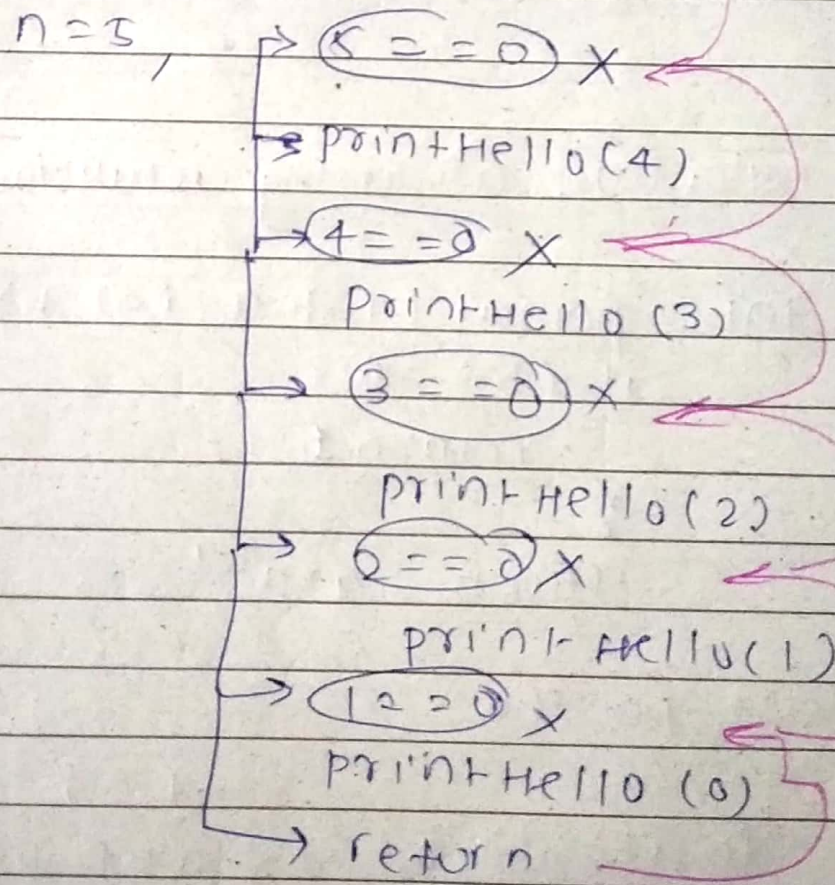
when a func^{n} calls itself.

eg:

```
void main(){
    printHello(5);
}

void printHello(int n){
        if(n==0){
            return;
        }
        printHello(n-1);
    }
```

Dry run.



3 steps for Recursion.

(i) find the best case
(ii) Find the relation between the problem & subproblem.
(iii) Generalise the relation.

eg2 Sum of n natural numbers using recursion

```
int sum(int n){
    if(n==1)
        return 1;
    return n + sum(n-1);
}
```

base case

eg3 calculate a^b using recursion

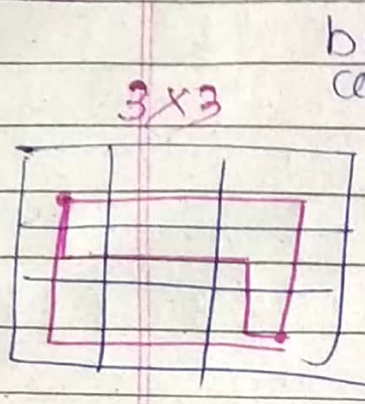
```
int power(int a, int b){
    if (b==0){
        return 1;
    }
    return a*a^(b-1);
}
```

⇒ fast power

$$fastPow(a,b) = \begin{cases} 1, & b = 0 \\ fastPow(a^2, b/2), & b \text{ is even} \\ a * fastPow(a, b-1), & b \text{ is odd} \end{cases}$$

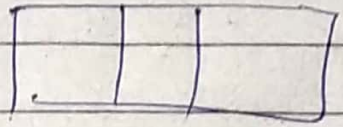
eg3 Find all the paths in a nxm grid.

base case ⇒ n = 1, m = 1

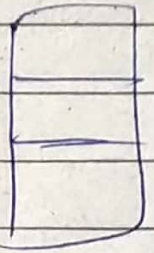


3×3

n = 1, m = 3



m = 1, n = 3



```
static int path (int n, int m) {
    if (n == 1 || m == 1)
        return 1;
    return path (n, m-1) + path (m, n-1);
}
```