L27
Recursion Basics

The System Design Course Offer ends tonight.
If interested, check out your whatsapp and
*purchase **before midnight.***

Messages were sent on 17th Feb, around 6:35PM.

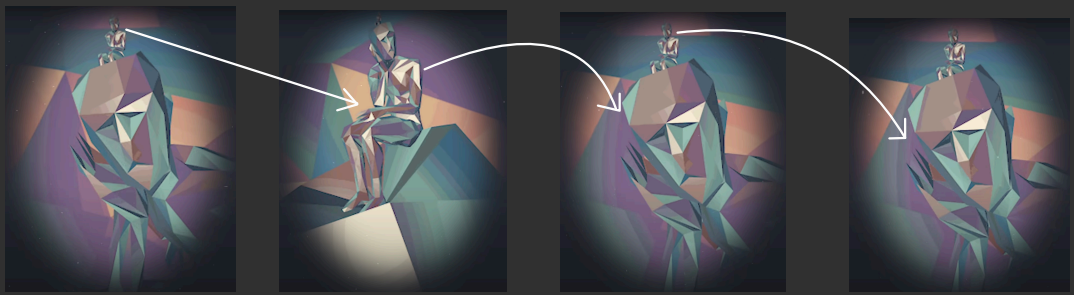
Join Discord - https://bit.ly/ly-discord

RECAP

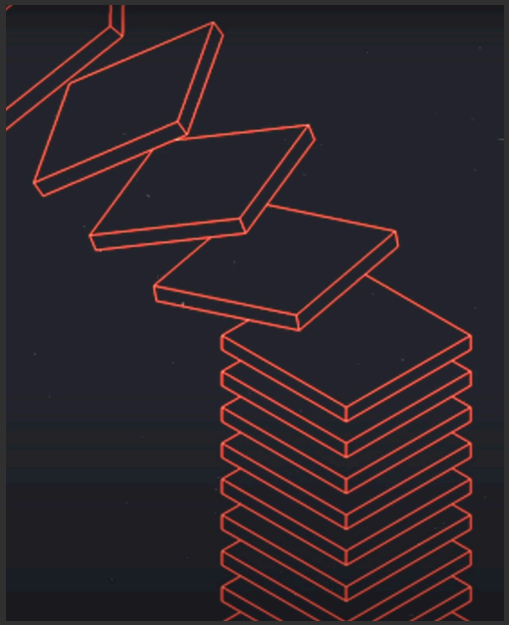
What is Recursion?

It’s magic

Imagine having a dream inside of a dream inside of a dream.
Pretty dreamy, right?

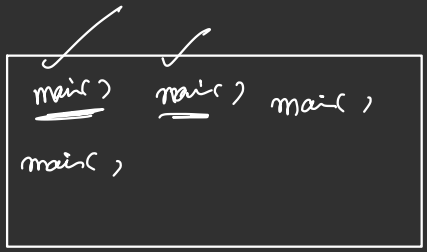


```
void dream() {
dream()
}
```

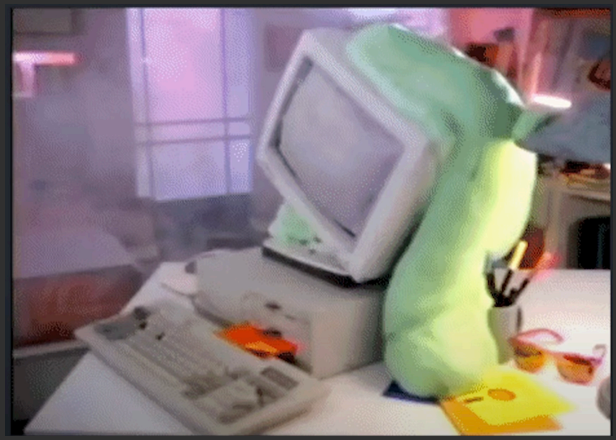


In layman terms, when a function calls itself, it’s called recursion.

```
main()
{
    print("Fsaz");
    main()
}
```



But, it just running indefinitely is of no use, right?



So, there needs to be a stopping condition
(AKA Base Condition)

```
void dream() {
if(woken_up)
   return;
dream()
}
```

```
int factorial (int n)
{ if (n==0) return 1;
  int partial_ans = factorial(n-1);
  int ans = partial_ans * n;
  return ans;
}
```

(5) (4) (1)

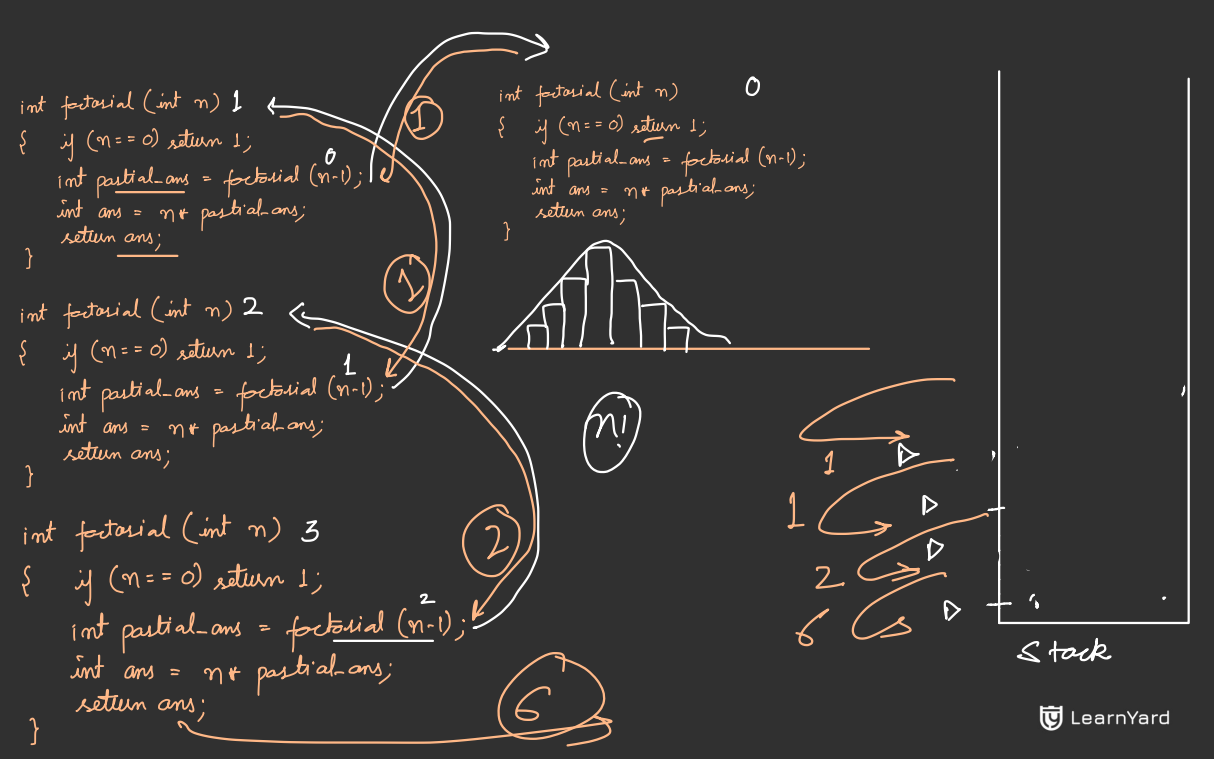
Let's try finding the factorial of a number

$5! = 1 \times 2 \times 3 \times 4 \times 5$

$5! = 4! \times 5$

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

How is it working internally?
AKA Call Stack

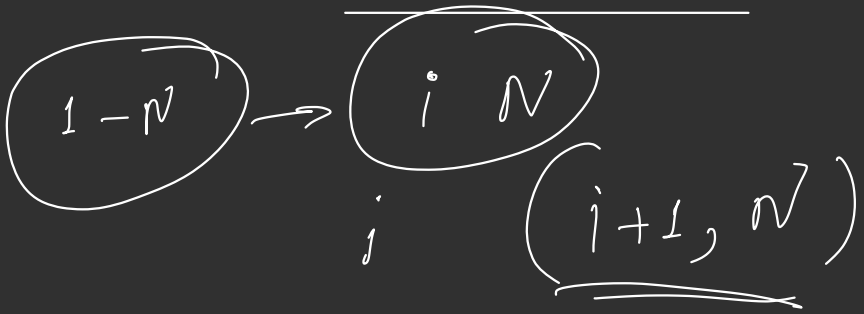

int factorial (int n) 1
{ if (n == 0) return 1;
int partial_ans = factorial (n-1);
int ans = n * partial_ans;
return ans;
}
0
int factorial (int n) 2
{ if (n == 0) return 1;
int partial_ans = factorial (n-1);
int ans = n * partial_ans;
return ans;
}
1
int factorial (int n) 3
{ if (n == 0) return 1;
int partial_ans = factorial (n-1);
int ans = n * partial_ans;
return ans;
}
2
int factorial (int n) 0
{ if (n == 0) return 1;
int partial_ans = factorial (n-1);
int ans = n * partial_ans;
return ans;
}
1
1
2
6
n!
Stack
LearnYard

Let’s do some more practice now.

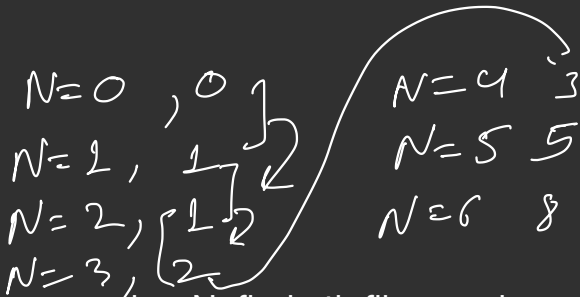
```
void printN(int n, int i)
{ if(i == n+1) return;
  print(i);
  printN(n, i+1);
}
```

```
main()
{ printN(5,1);
```

Given a value of N, print numbers from 1 to N.

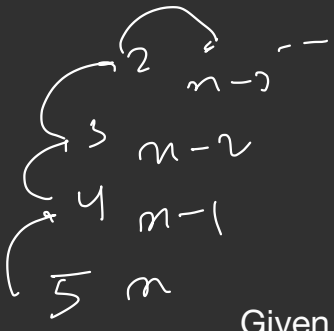


Given a value of N, print numbers from N to 1.



Given a number N, find nth fibonacci number (f0 = 0, f1 = 1)
(Fn = Fn-1 + Fn-2)

```
int fibo(n) {
    if (n==0 || n==1) return n;
    int a = fibo(n-1);
    int b = fibo(n-2);
    return a+b;
}
```



$2^{-4} = \frac{1}{2^4}$

$2^{-n} = \frac{1}{2^n}$

Given a number a, n - find pow(a,n)

a^n

$x^n \rightarrow x * x^{n-1}$

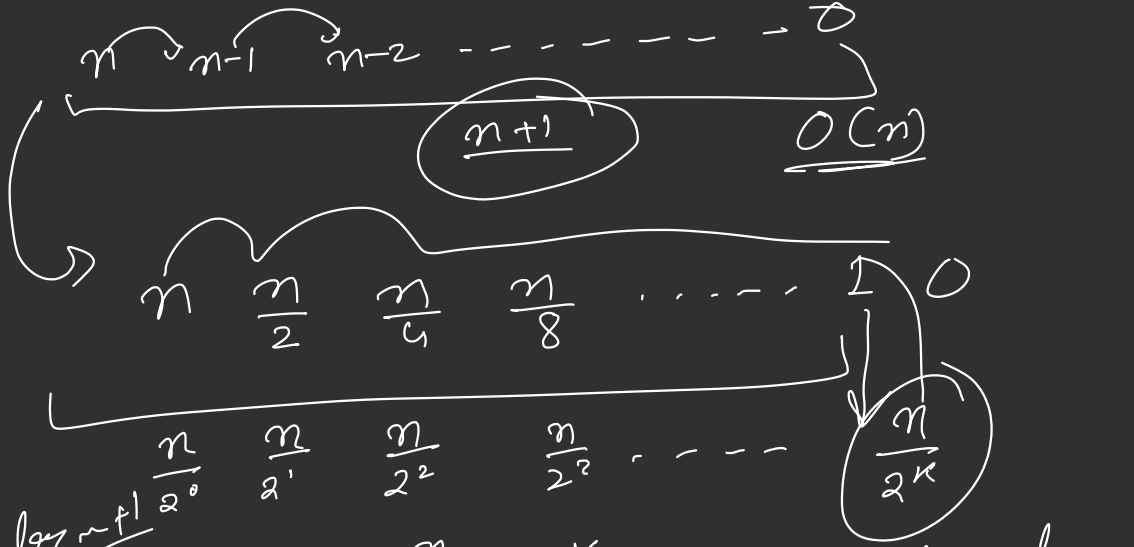
$x^n \rightarrow x^{n/2} * x^{n/2}$

$x^6 \rightarrow x^3 * x^3$

$x^7 \rightarrow x^3 * x^3 * x$

$x^n \rightarrow x^{n/2} * x^{n/2}$ n is even

$\rightarrow x^{n/2} * x^{n/2} * x$ n is odd



$$1 = \frac{n}{2^k} \qquad 2^k = n$$

$$\log_2 2^k = \log n$$

$$k \log 2 = \log n$$

Given a number n, check if n is a perfect power of 3.

Thank You!

Reminder: Going to the gym & observing the trainer work out can help you know the right technique, but you'll muscle up only if you lift some weights yourself.

So, PRACTICE, PRACTICE, PRACTICE!