



-Intro to recursion

- ✓ - How to write recursive code
- ✓ - sum of n
- ✓ - factorial n!
- ✓ - tracing
 - Fibonacci
 - print 1-n
- check palindrome

TC / SC of recursive codes next session

why Recursion?

- Merge Sort, Quick sort

- Binary Tree, BST, segment tree , tries

- Dynamic Programming

- Backtracking

- Graphs

Recursion? function that calls itself

Solving problem using smaller instance of
the problem
sub problem

$$\text{sum}(4) = 4 + \underbrace{3+2+1}_{\text{sum}(3)}$$

recursive
form

$$\text{sum}(n) = n + \text{sum}(n-1)$$

How to write recursive code?

- ① Assumption decide what your func. does
- ② Main Logic solve problem using subproblems
- ③ Base Condition Input for which we need stop

sum of n natural numbers

ex int sum(n) {
 if(n == 1) ret 1
 ret n + sum(n-1)
}

✓① Assumption

✓② Main Logic

✓③ Base Condition

-1

Quiz ex

$$\text{fact}(n) = n \times (n-1) \times \dots \times 1$$

$$4! = 4 \times 3 \times 2 \times 1 = 24$$

int fact(n){
 if(n == 1) ret 1
 ret n * fact(n-1)
}

↳ fact(n-1)

$$6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1$$

$$1! = 1$$

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

24 and

✓① Assumption

✓② Main Logic

✓③ Base Condition

function int add(a, b){

 ret a+b

tracing }

main(){

 x=10, y=20 825

 print(sub(mul(add(x,y), 30), 75))

 int mul(x, y){

 ret x*y

}

 } ↳ sub(mul(add(x,y), 30), 75)

 int sub(P, Q){

 ret P-Q

}

 int div(a, b){

 ret a/b

}

 mul(add(x,y), 30) = 900

 sub(900, 75)

 725

 add(x,y)

 10+20 = 30

output = 825

nested function calls

Related data structure

$$\cancel{\text{add}(x,y) = 10+20 \approx 30}$$

$$\cancel{\text{mul}(\text{add}(x,y), 30)}$$

$$\cancel{\text{sub}(\text{mul}(\text{add}(x,y), 30), 75)}$$

$$\cancel{\text{print}(\text{sub}(\text{mul}(\text{add}(x,y), 30), 75))}$$



Stack

① we add on top

② we remove from top

$$\begin{array}{r} 4+ \\ 3+ \\ 2+ \\ \hline 1 = 10 \end{array}$$

Sum Tracing

```
int sum(4){  
    if(n>=1) ret 1  
    → ret sum(n-1)+n  
}
```

```
int sum(3){  
    if(n>=1) ret 1  
    → ret sum(n-1)+n  
}
```

```
int sum(2){  
    if(n>=1) ret 1  
    → ret sum(n-1)+n  
}
```

```
int sum(1){  
    if(n>=1) ret 1  
    ret sum(n-1)+n  
}
```

~~sum(1)~~
~~sum(2)~~
~~sum(3)~~
~~sum(4)~~

$$\text{if } n \geq 100 \text{ ret } \frac{100 \times 101}{2}$$

Stack overflow

Base condition issues

pro tip: if I have rec. call and get MLE → check first base case

$n \geq 0$

Fibonacci	0	1	2	3	4	5	6	7	8	9	10
numbers	0	1	1	2	3	5	8	13	21	34	...

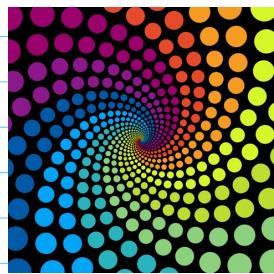
Quiz

- ✓ ① Assumption
- ✓ ② Main Logic
- ✓ ③ Base condition

```
int Fib(n){
    if(n==0) ret 0 } if(n<=1) ret n
    if(n>1) ret 1 }
    ret Fib(n-1)+Fib(n-2)
}
```

$\text{if}(n \geq 0) \quad F(2)$

$$F(7) = F(6) + F(5)$$



trace
for
Fib

$n=4$

```
int Fib( ){  
    if(n==0) ret 0  
    if(n==1) ret 1  
    ret Fib(n-1)+Fib(n-2)  
}
```

```
int Fib(4){  
    if(n==0) ret 0  
    if(n==1) ret 1  
    → ret Fib(n-1)+Fib(n-2)  
}
```

```
int Fib(3){  
    if(n==0) ret 0  
    if(n==1) ret 1  
    ret Fib(n-1)+Fib(n-2)  
}
```

```
int Fib(2){  
    if(n==0) ret 0  
    if(n==1) ret 1  
    ret Fib(n-1)+Fib(n-2)  
}
```

```
int Fib(2){  
    if(n==0) ret 0  
    if(n==1) ret 1  
    ret Fib(n-1)+Fib(n-2)  
}
```

```
int Fib(1){  
    if(n==0) ret 0  
    if(n==1) ret 1  
    ret Fib(n-1)+Fib(n-2)  
}
```

```
int Fib(1){  
    if(n==0) ret 0  
    if(n==1) ret 1  
    ret Fib(n-1)+Fib(n-2)  
}
```

```
int Fib(0){  
    if(n==0) ret 0  
    if(n==1) ret 1  
    ret Fib(n-1)+Fib(n-2)  
}
```

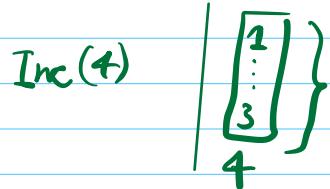
```
int Fib(1){  
    if(n==0) ret 0  
    if(n==1) ret 1  
    ret Fib(n-1)+Fib(n-2)  
}
```

```
int Fib(0){  
    if(n==0) ret 0  
    if(n==1) ret 1  
    ret Fib(n-1)+Fib(n-2)  
}
```

P1 Given N, print all numbers from 1 to N in increasing order; use recursive solutions

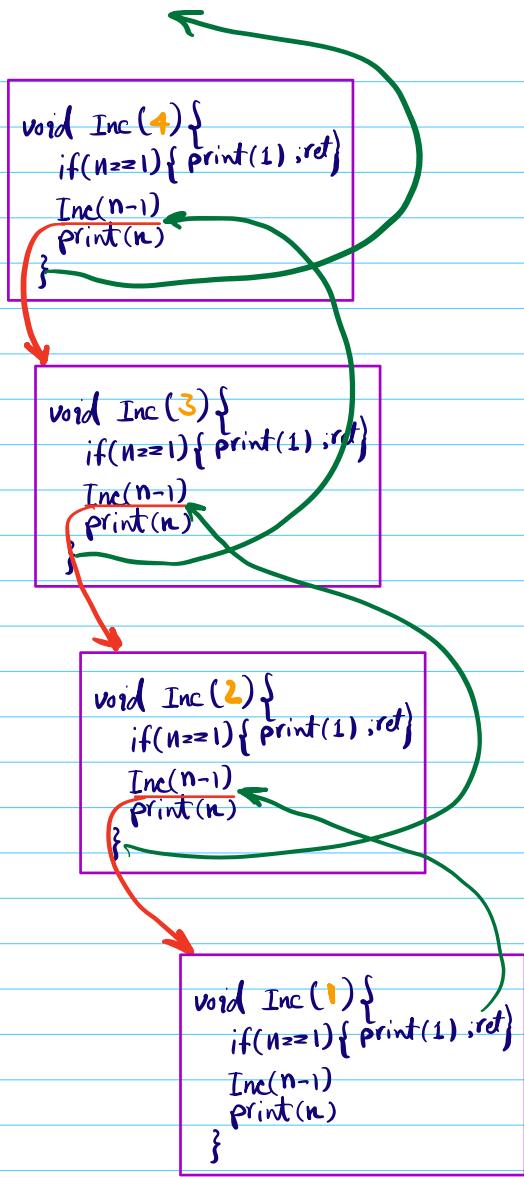
```
void Inc(n){  
    if(n>=1){ print(1); ret }  
    Inc(n-1)  
    print(n)  
}
```

- ✓① Assumption
- ✓② Main Logic
- ✓③ Base Condition



Optional HW: swap Inc(n-1) and Print lines
and observe output ...
what does this new code do?

trace
print
 $n=4$



true/false

P2 Given a substring, check if it is palindrome or not.

ex "gooddad" true

vaor x
moon x

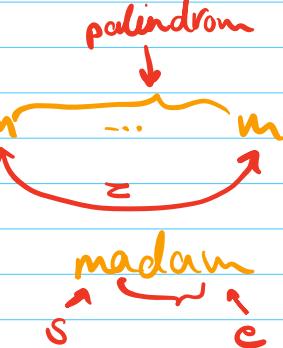
"gooddad" false xyzmadam

sub

Quiz &
Quiz

- ✓ ① Assumption
- ✓ ② Main Logic
- ③ Base condition

```
bool isPalindrome(char ch[], int s, int e) {
    if (s >= e) ret true
    {if(ch[s] == ch[e] && isPalindrome(ch,s+1,e-1)){
        ret true
    } else ret false
}
```



Can be rewritten

```
if(ch[s] == ch[e]){
    ret isPalindrome(ch,s+1,e-1)
}
```

X Y Y X
e s

ans true
trace 0 1 2 3 4 5 6 7
for XXZMAADAM

Input

```
bool isPalindrome(char ch[], int s, int e){  
    if (s>e) ret true  
    if (ch[s]==ch[e] &&  
        isPalindrome(ch, s+1, e-1))  
        ret true  
    } else ret false  
}
```

```
bool isPalindrome(char ch[], int s, int e){  
    if (s>e) ret true  
    if (ch[s]==ch[e] &&  
        isPalindrome(ch, s+1, e-1))  
        ret true  
    } else ret false  
}
```

```
bool isPalindrome(char ch[], int s, int e){  
    if (s>e) ret true ✓ 5 5  
    if (ch[s]==ch[e] &&  
        isPalindrome(ch, s+1, e-1))  
        ret true  
    } else ret false  
}
```

ang **false**
0 1 2 3 **↑**
M A D X M

```
bool isPalindrome(char ch[], int s, int e){  
    if (s>=e) ret true  
    if (ch[s]==ch[e] &&  
        isPalindrome(ch, s+1, e-1))  
        ret true  
    }  
    else ret false  
}
```

```
bool isPalindrome(char ch[], int s, int e){  
    if (s>=e) ret true  
    if (ch[s]==ch[e] &&  
        isPalindrome(ch, s+1, e-1))  
        ret true  
    }  
    else ret false  
}
```