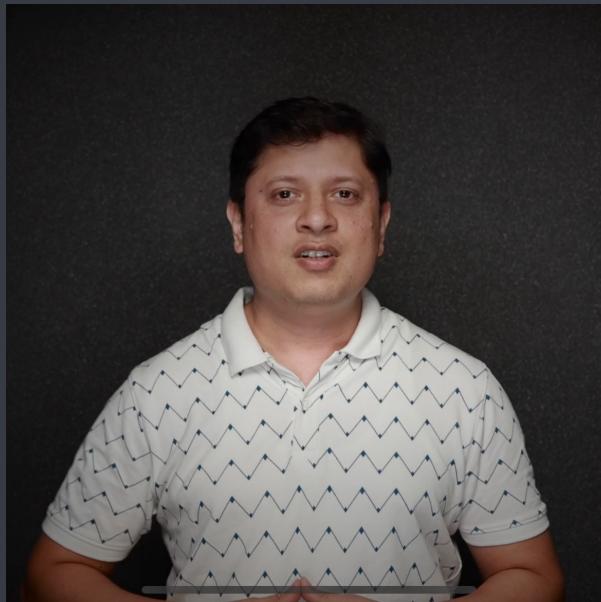


# C Language

## Recursion



Saurabh Shukla (MySirG)

## Agenda

- ① What is a recursion?
- ② Recursion Tree | Tracing code
- ③ How to approach recursive solution?
- ④ Few examples

## What is a recursion?

- Function calling itself is called recursion
- A recursive method solves a problem by calling a copy of itself to work on a smaller problem.
- It is important to ensure that the recursion terminates.

```

void f1()
{
    printf("Hello"),
    f2(); f1();
    printf("Bye");
}

```

```

void f2()
{
    printf("A");
}

```

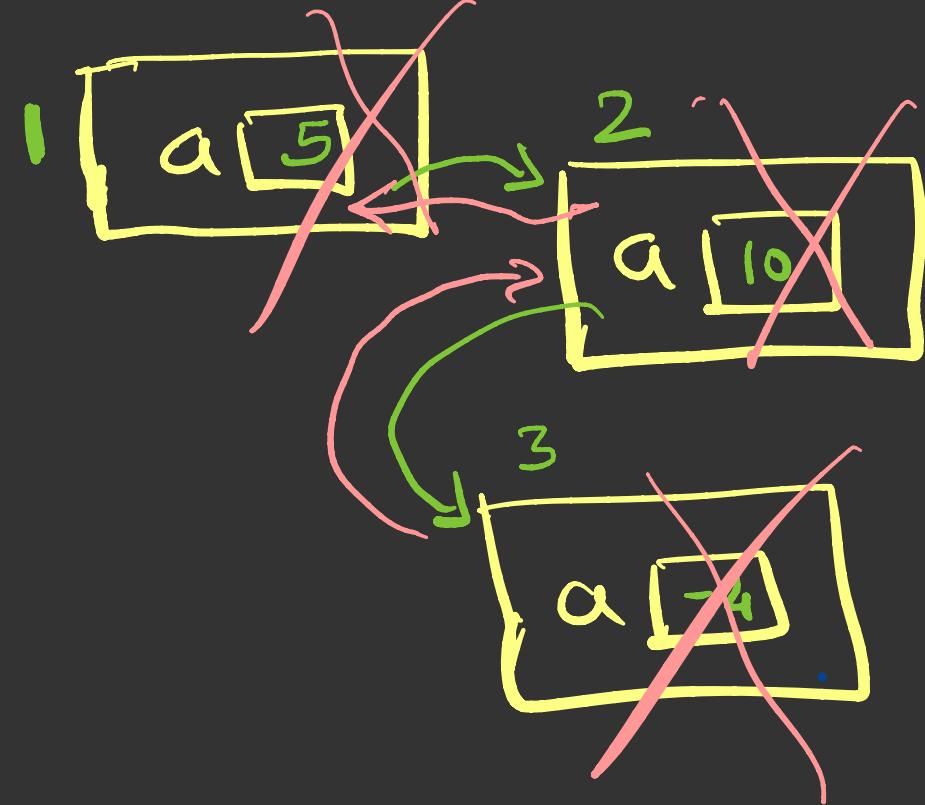
$\vdash$  f1()    pf(Hello)    Hello  
 $\vdash$  f1() — E f1()    pf(Hello)    Hello  
 $\vdash$  f1() — E f1() — E f1()    pf(Hello)    Hello  
 $\vdash$  f1() — E f1() — E f1() — E f1()    pf(Hello)    Hello

```

void f1()
{
    int a;
    printf("Enter a number");
    scanf("%d", &a);
    if(a>0)
        f1();
    printf("%d", a);
}

```

Enter a number 5  
 Enter a number 10  
 Enter a number -4  
 -4 10 5



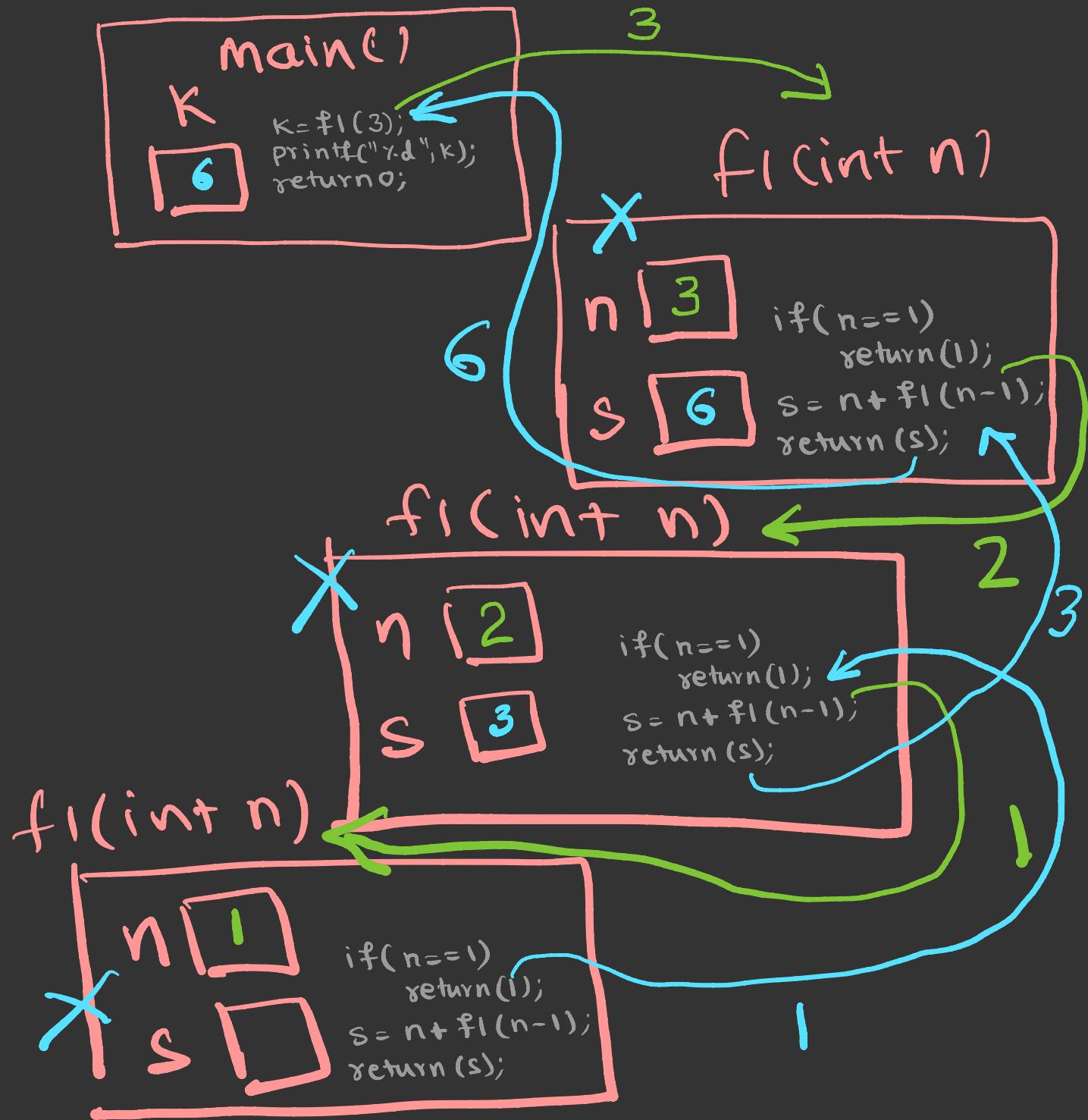
- Each time the function call itself with a slightly simpler version of the original problem.
- Recursive code is generally shorter and easier to write than iterative code.
- Solution to some problems are easier to formulate recursively.

```

int main()
{
    int K;
    K=f1(3);
    printf("y.d",K);
    return 0;
}

int f1( int n)
{
    int s;
    if( n==1)
        return(1);
    s= n+f1(n-1);
    return (s);
}

```



$$f_1(1) = 1$$

$$f_1(2) = 3$$

$$f_1(3) = 6$$

$$f_1(4) = 10$$

$$f_1(5) = 15$$

$$f_1(20)$$

$$f_1(n)$$

$$2 + f_1(1) \quad 2+1$$

$$3 + f_1(2) \quad 3+2+1$$

$$4 + f_1(3) \quad 4+3+2+1$$

$$5 + f_1(4) \quad 5+4+3+2+1$$

$$20 + f_1(19) \quad \begin{matrix} 20+19+.. \\ \dots 3+2+1 \end{matrix}$$

$$n + f_1(n-1)$$

$$\begin{matrix} n+(n-1)+(n-2)+.. \\ \dots 3+2+1 \end{matrix}$$

## How to approach a Recursive Problem?

Write a recursive function to calculate sum of first n natural numbers.

- ①  $\text{sum}(n) \quad 1+2+3+\dots+n$
- ②  $n + \text{sum}(n-1) \quad 1+2+3+\dots+n-1$
- ③  $\text{sum}(1) \quad 1$

int sum(int n)

{

    if ( $n == 1$ )

        return 1;

    return  $n + \text{sum}(n-1);$

}