

C Language LIVE Community Classes

Recursion

Day-15

By
Saurabh Shukla
(MySirG)

What is 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.

- Each time the function call itself with a slightly simpler version of the original problem.

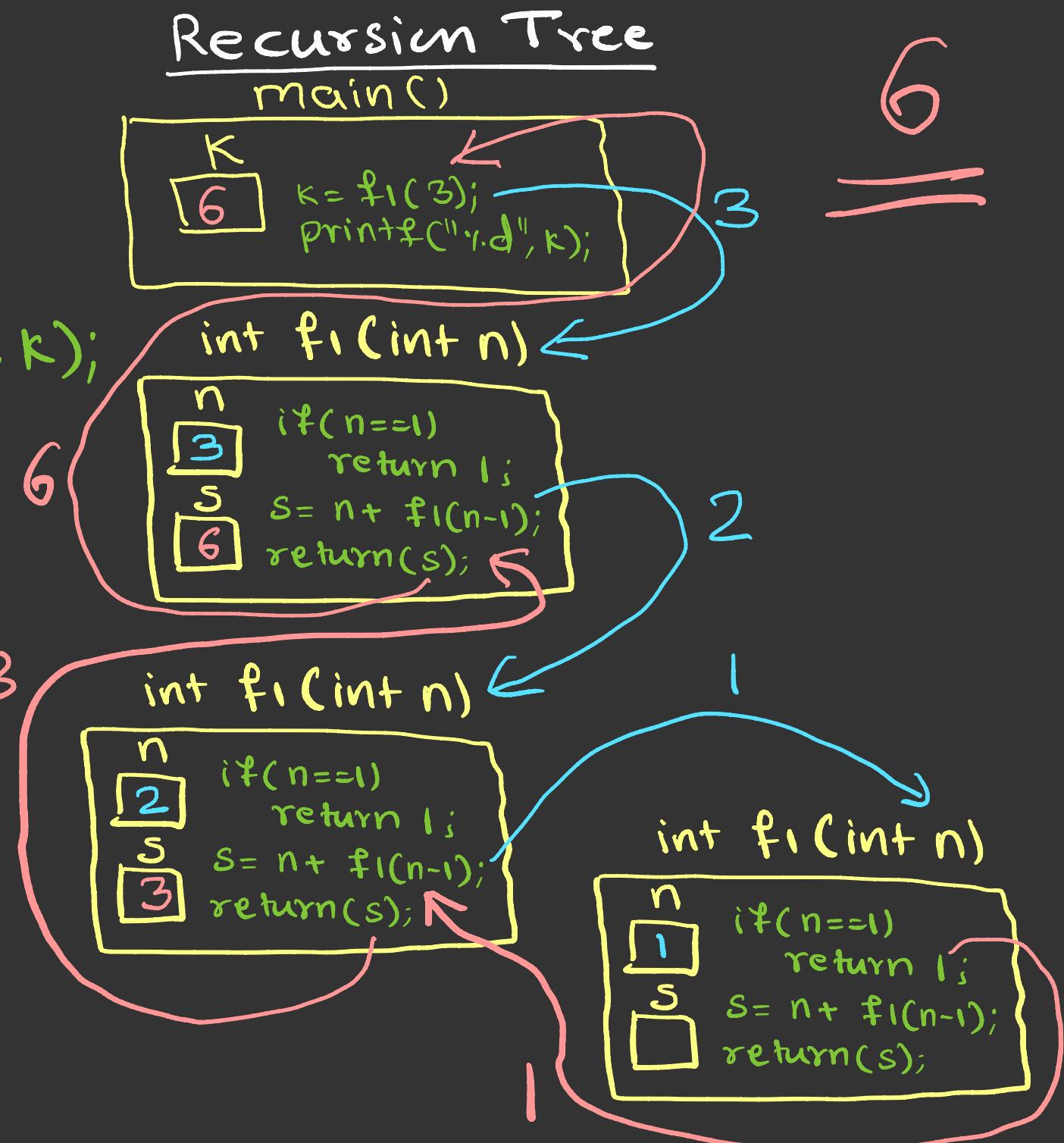
- Recursive code is generally shorter and easier to write than iterative code.
- Solution to some problems are easier to formulate recursively.

```

int main()
{
    int K;
    K = f1(3);
    printf("%d", K);
}

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

```



$$K = \cancel{f_1(10)}$$

$$10 + \cancel{f_1(9)}$$

$$9 + \cancel{f_1(8)}$$

$$8 + \cancel{f_1(7)}$$

$$7 + \cancel{f_1(6)}$$

$$6 + \cancel{f_1(5)}$$

$$5 + \cancel{f_1(4)}$$

$$4 + \cancel{f_1(3)}$$

$$3 + \cancel{f_1(2)}$$

$$2 + \cancel{f_1(1)}$$

$$K = \underline{\cancel{f_1(3)}}$$

$$3 + f_1(2)$$

$$\begin{matrix} 2 + f_1(1) \\ \uparrow \\ 1 \end{matrix}$$

$$K = 3 + 2 + 1$$

$$f_1(n)$$

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

Base case

$$K = 10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 = 55$$

How to approach a Recursive Problem?

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

```
int sum(int n)
```

```
{
```

```
    if(n==1)
```

```
        return 1;
```

```
    return (n + sum(n-1));
```

```
}
```

- ① Sum(n) $1+2+3+4+\dots+n$
- RC ② $n + \text{sum}(n-1)$ $1+2+3+\dots+n-1$
- BC ③ $n = 1$ return 1

Write a recursive function to calculate factorial of n.

```
int fact(int n) {  
    if (n == 0)  
        return 1;  
    return n * fact(n-1);  
}
```

① fact(n) $1 * 2 * 3 * \dots * n$
② ^{RC} $n * \text{fact}(n-1)$ $1 * 2 * 3 * \dots * n-1$
③ ^{BC} $n == 0$ return 1

$1! = 1$
 $0! = 1$
 $2! = 2 * 1$
 $3! = 3 * 2 * 1$

Write a recursive function to print first n natural numbers.

```
void printN(int n) {  
    if(n>0) {  
        printN(n-1);  
        printf("%d", n);  
    }  
}
```

① printN(n) | 2 3 4 ... n
RC ② printN(n-1) | 2 3 ... n-1
printf("%d", n); n
③ n == 0