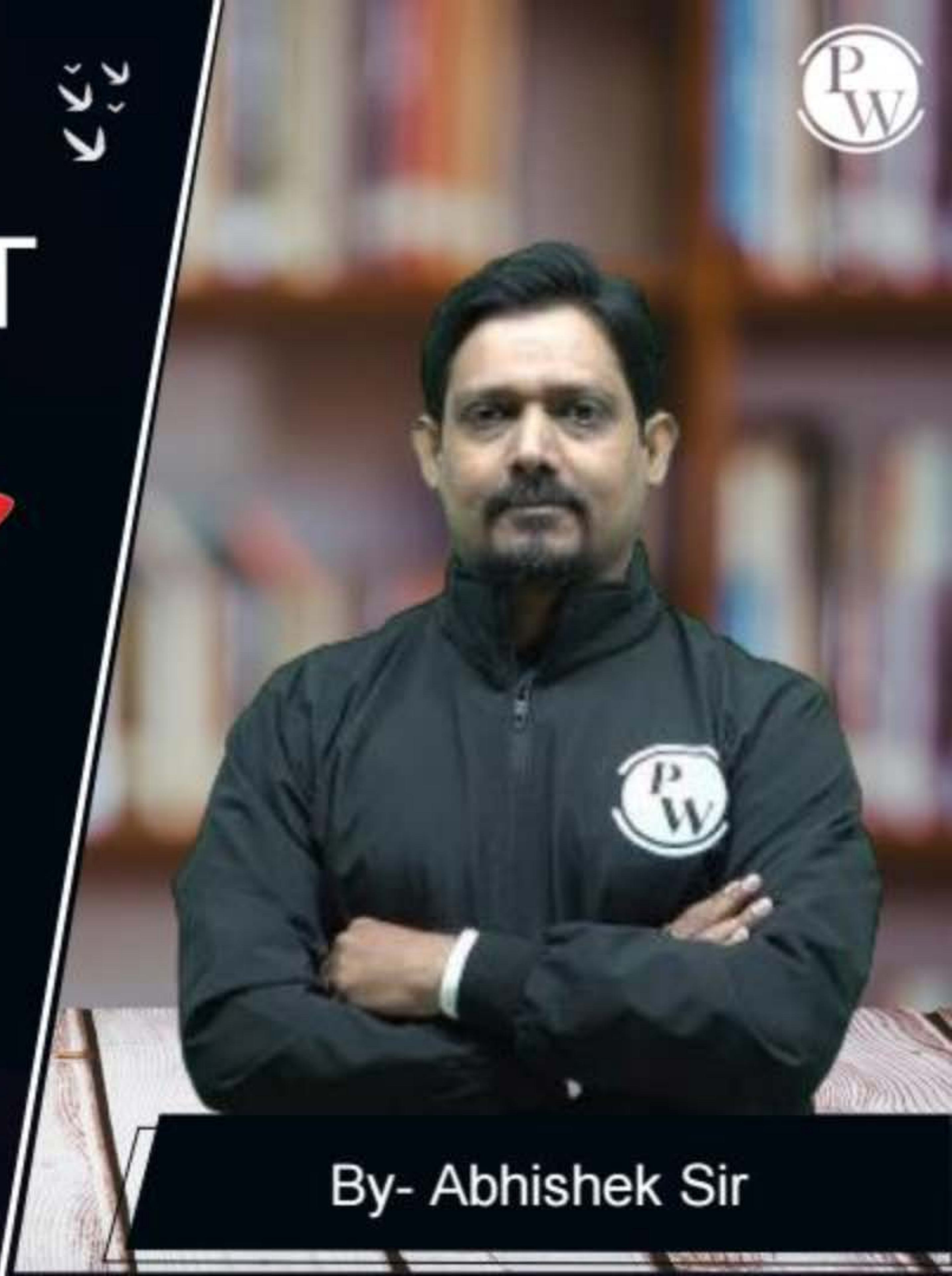
Computer Science & IT

C Programming

Function & Storage Class

Lecture No. 03



By- Abhishek Sir

Recap of Previous Lecture

Topic

Static variable

Topic

properties of static variable (Imp)

Topic

Topic

Topic

Slide

Topics to be Covered

Topic Recursion

Topic Type of Recursion

Topic Activation Tree.

Topic

Topic

Slide

Question

#Q Consider the following C function:

```
int f(int n){
        static int i = 1 ;
        if (n >=5)
                return n;
        n = n+i;
        i++;
        return f(n);
}
```

$\uparrow$ Recursion

The value returned by $\underline{f(1)}$ is

(a) 5
(b) 6
(c) 7
(d) 8

Slide

GATE 2004

#Q

$y = y + foo(1) + bar(1)$

```
#include <stdio.h>
int foo(int x);
int bar(int y);
int main ()       {
        int x = 1, y = 2, count;
        for (count = 1; count <=2; ++count){
                y+=foo(x) + bar(x);
        printf("\n %d",y);
        }
}
```

```
int foo (int x) {
        int y;
        y=bar(x);
        return(y);
}
int bar (int x) {
        static int y = 10;
        y+=1;
        return (y+x);
}
```

$x = 1$

$y = bar(\overset{1}{x})$

$y = 12$

$y = \cancel{10}\ \cancel{11}\ 12$

$\cancel{y=}$

$11+1$

$12+1$

$y = 2 + \underset{12}{foo(\overset{1}{x})} + \underset{13}{bar(\overset{1}{x})}$ $\boxed{= 27}$

$y = y + foo(x) + bar(x)$

$27 + 14 + 15$

$27 +$

$x = 1$

$y = bar(1)$

$y = 14$

27
14
15
56

[✓ (A) 27 56] (B) 42 74 (C) 33 37 (D) 32 32

Slide

GATE 2004

#Q

```
#include <stdio.h>
int foo(int x);
int bar(int y);
int main ()      {
        int x = 1, y = 2, count;
        for (count = 1; count <=2; ++count){
                y+=foo(x) + bar(x);
        printf("\n %d",y);
        }
}
```

```
int foo (int x) {
        int y;
        y=bar(x);
        return(y);
}
int bar (int x) {
        static int y = 10;
        y+=1;
        return (y+x);
}
```

$y = y + foo(1) + bar(1)$

$2 \quad 12 + 13 = 27$

$y = bar(x)$

$y = 12$

$13+1=14$

$y = \cancel{12}\ 13$

$14+1=15$

$y = y + foo(1) + bar(1)$

$= 27 + 14 + 15$

$= \boxed{56}$

[✓ (A) 27 56]   (B) 42 74   (C) 33 37   (D) 32 32

Slide

Storage Class

1. Auto
2. static
3. Extern ← No gate question
4. Register ← 1 gate question

} Tuesday last class

Slide

Recursion

* Recursion is a problem Solving technique in which Solution of a problem is expression in terms of smaller instance of same problem.

* In C Language it takes a function that call itself.

Slide

Recursion

Recursion

factorial

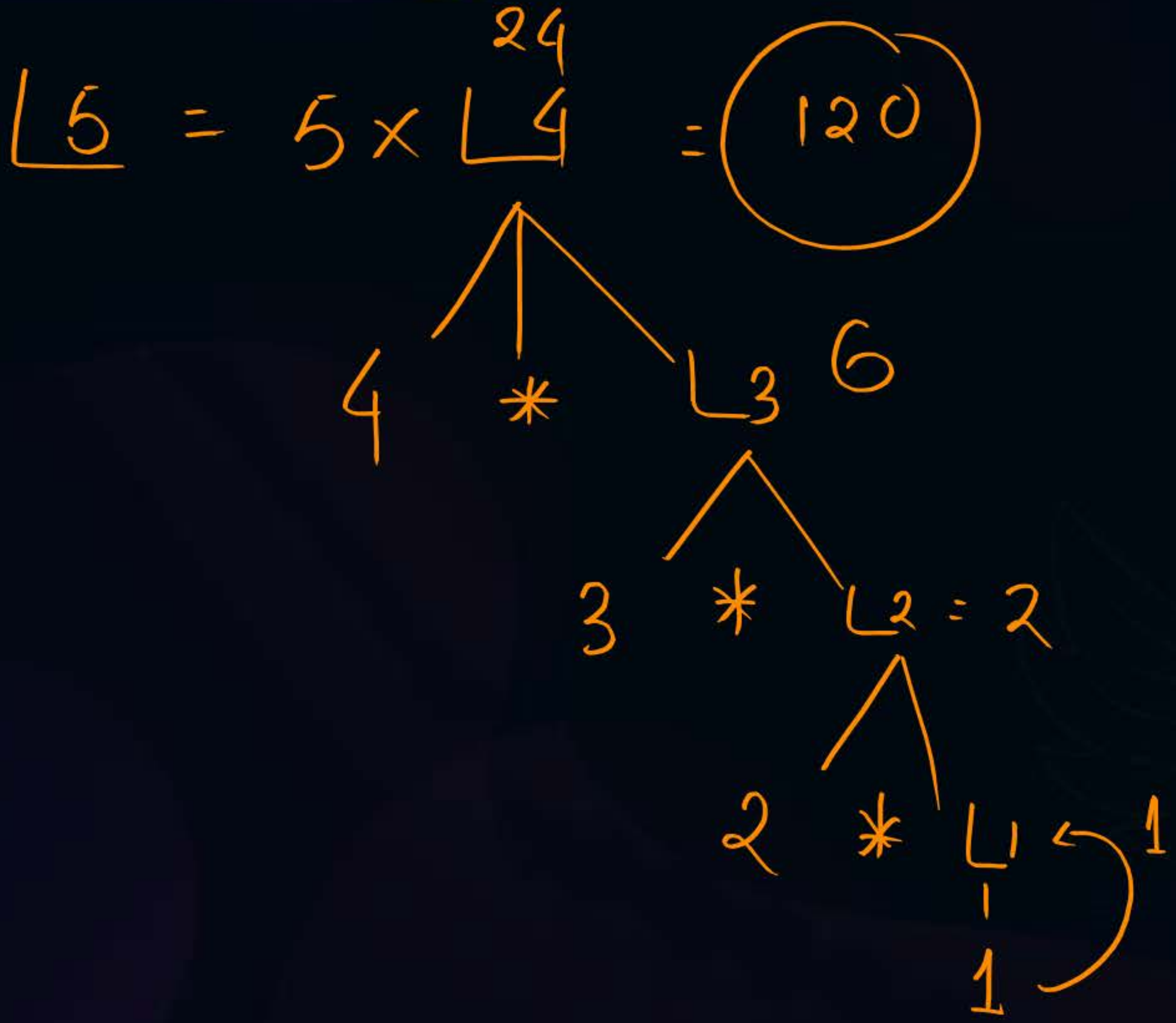
$$n! = n \times n-1!$$

$$\underline{|n} = \begin{cases} n \times \underline{|n-1} & n>1 \\ 1 & , n=0 \,||\, n=1 \end{cases}$$

Base Condition

Slide

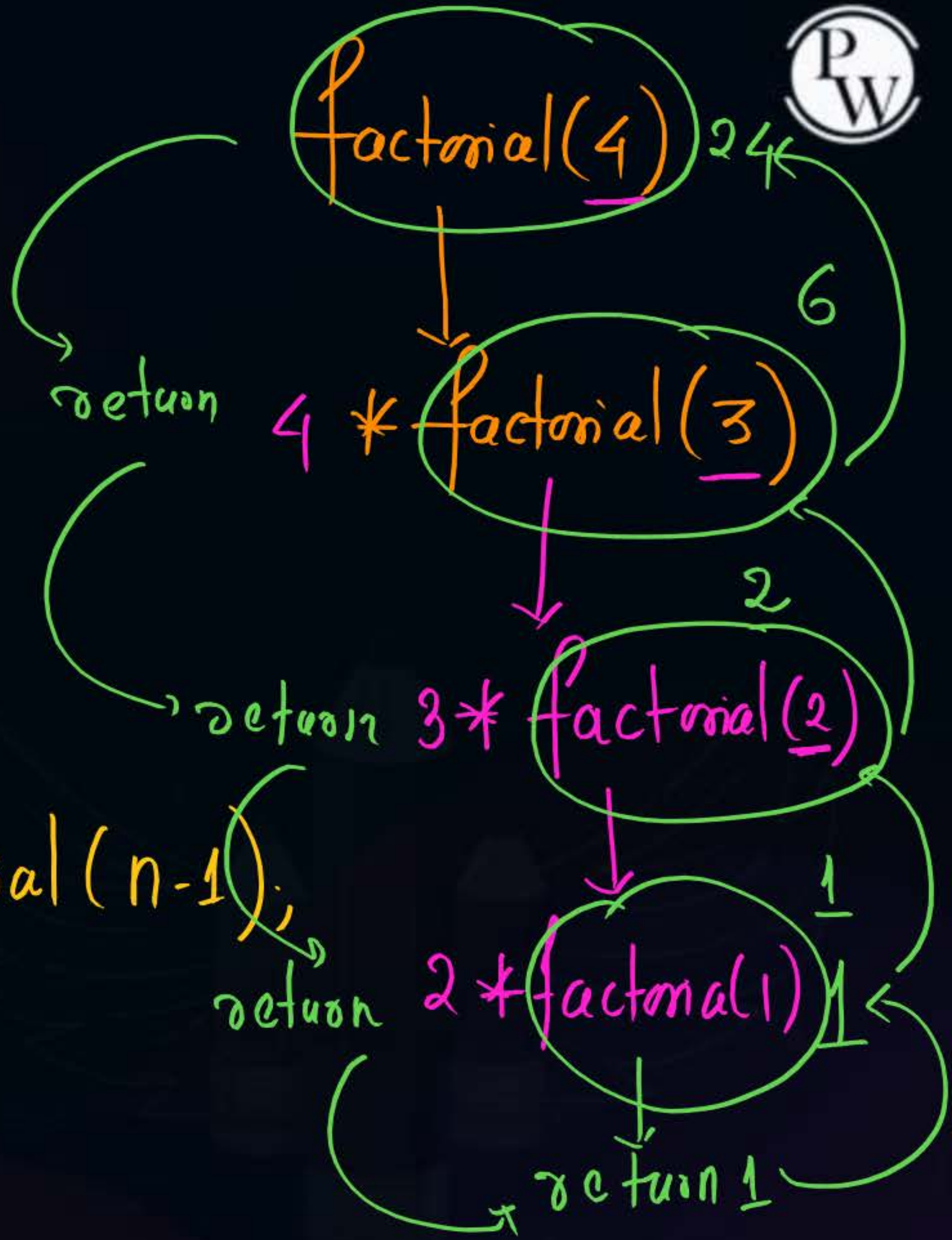
Recursion



Slide

Recursion

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



Slide

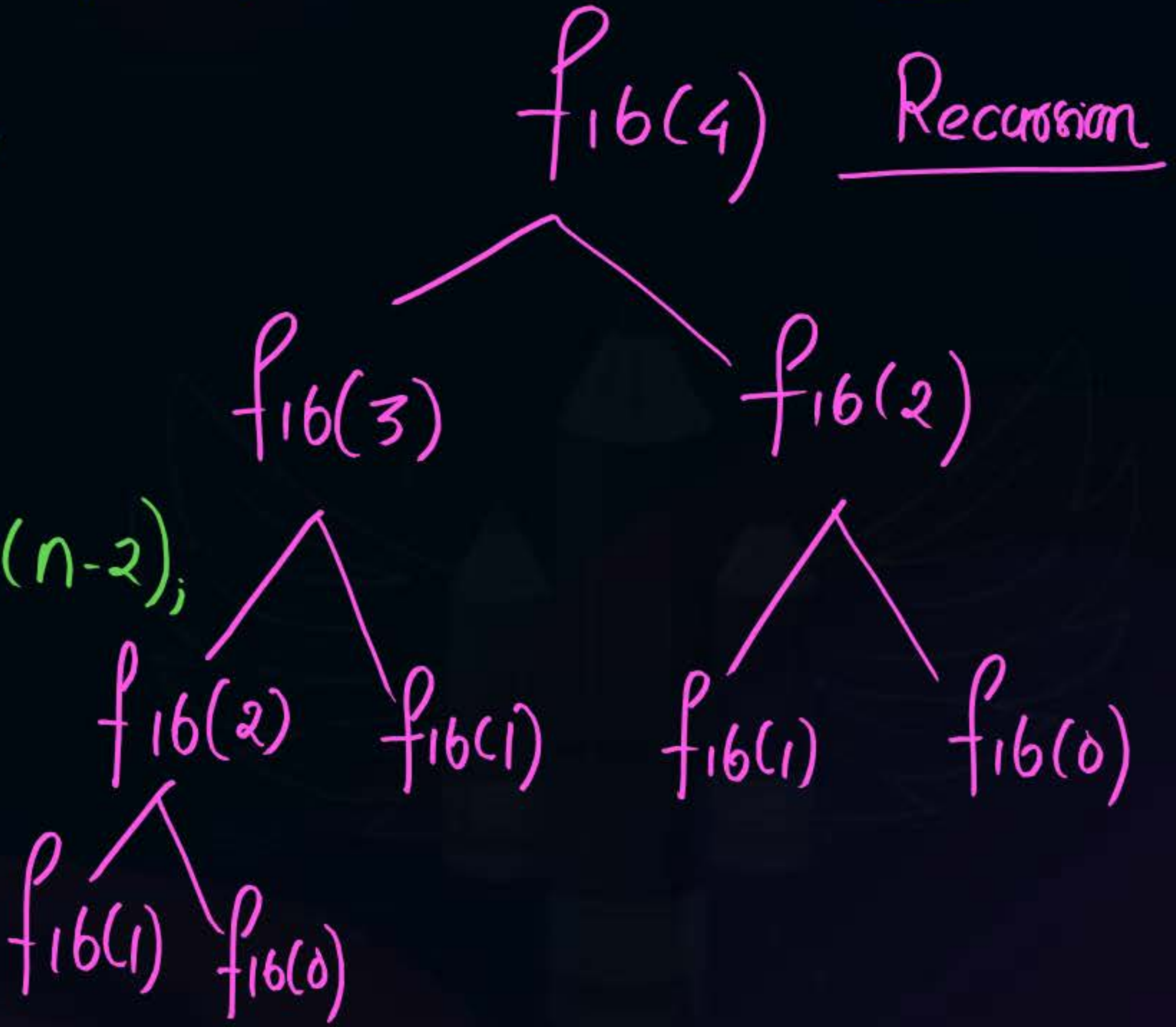
Recursion

Fibonacci Series

0, 1, 1, 2, 3, 5, 8, 13, 21 . . . . .

```
int fib(int n) {
    if(n==0||n==1) ←
        return n;
    else
        return fib(n-1)+fib(n-2);
}
```

fib(4) Recursion Tree draw



Slide

Recursion

```
int fib(int n) {
    if (n==0 || n==1)
        return n;
    else
        return fib(n-1) + fib(n-2);
}
```

HW

(1) The value of fib(13)

(2) The Number of times fib( ) function called during fib(8)

Slide

Recursion

1. value return
2. value printed (if given) ✓
3. No. of times function is called.

Slide

Recursion

Single Recursive call

1. Tail Recursion
2. Non tail Recursion

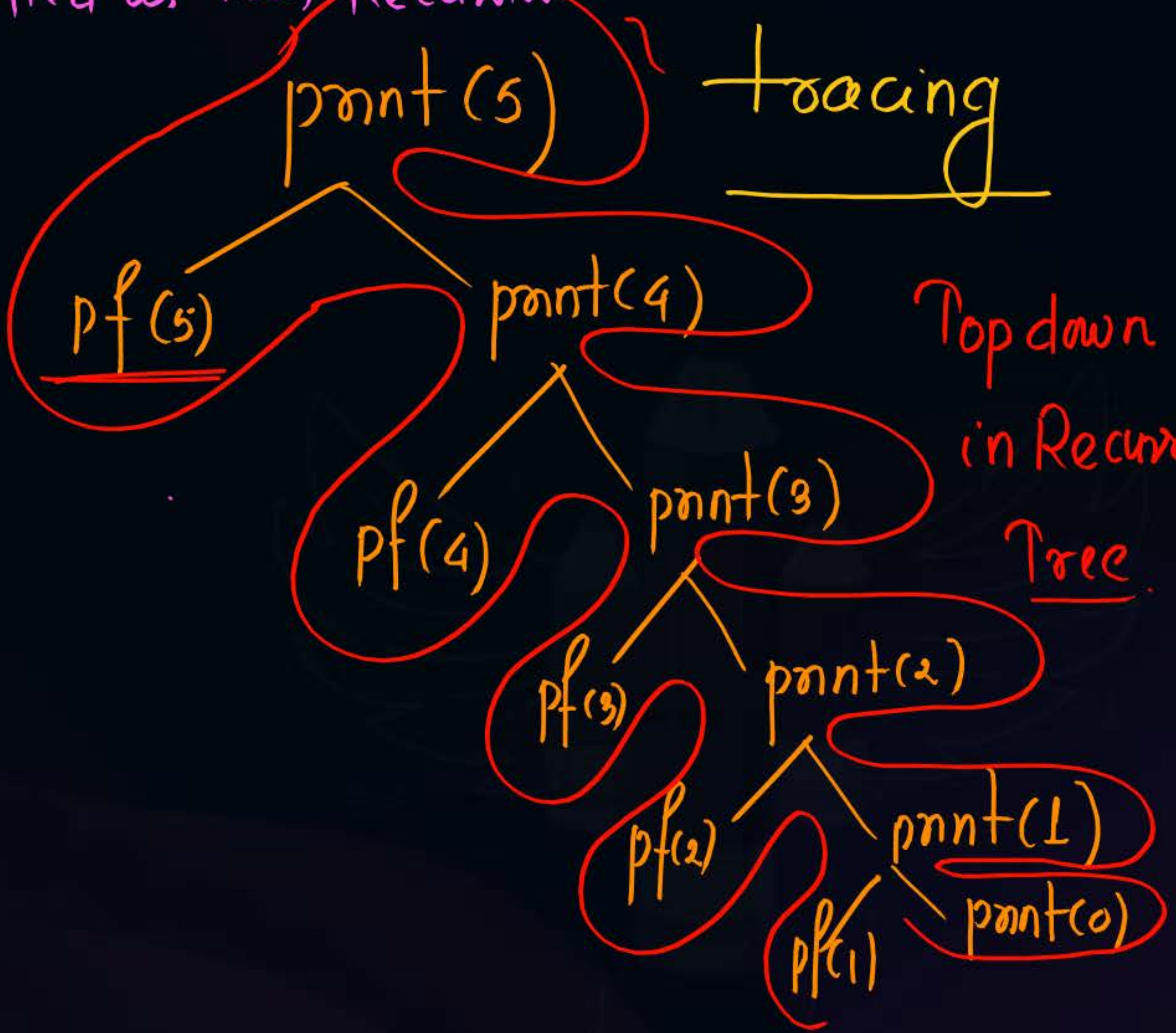
Slide

Recursion

if Recursive call is last statement of Recursion then its called as tail Recursion.

```
#inalcude<stdio.h>
void print(int n){
     if (n <= 0)  return; ✓
     printf("%d", n); ✓
     print(n-1); ✓
}
int main(){
     print(5); ✓
     return 0;
}
```

5, 4, 3, 2, 1



} tracing

Top down in Recursion Tree.

Slide

Recursion

Non Tail (Head Recursion)

Recursive call is first statement to execute

Recursion Tree and trace it & print output.

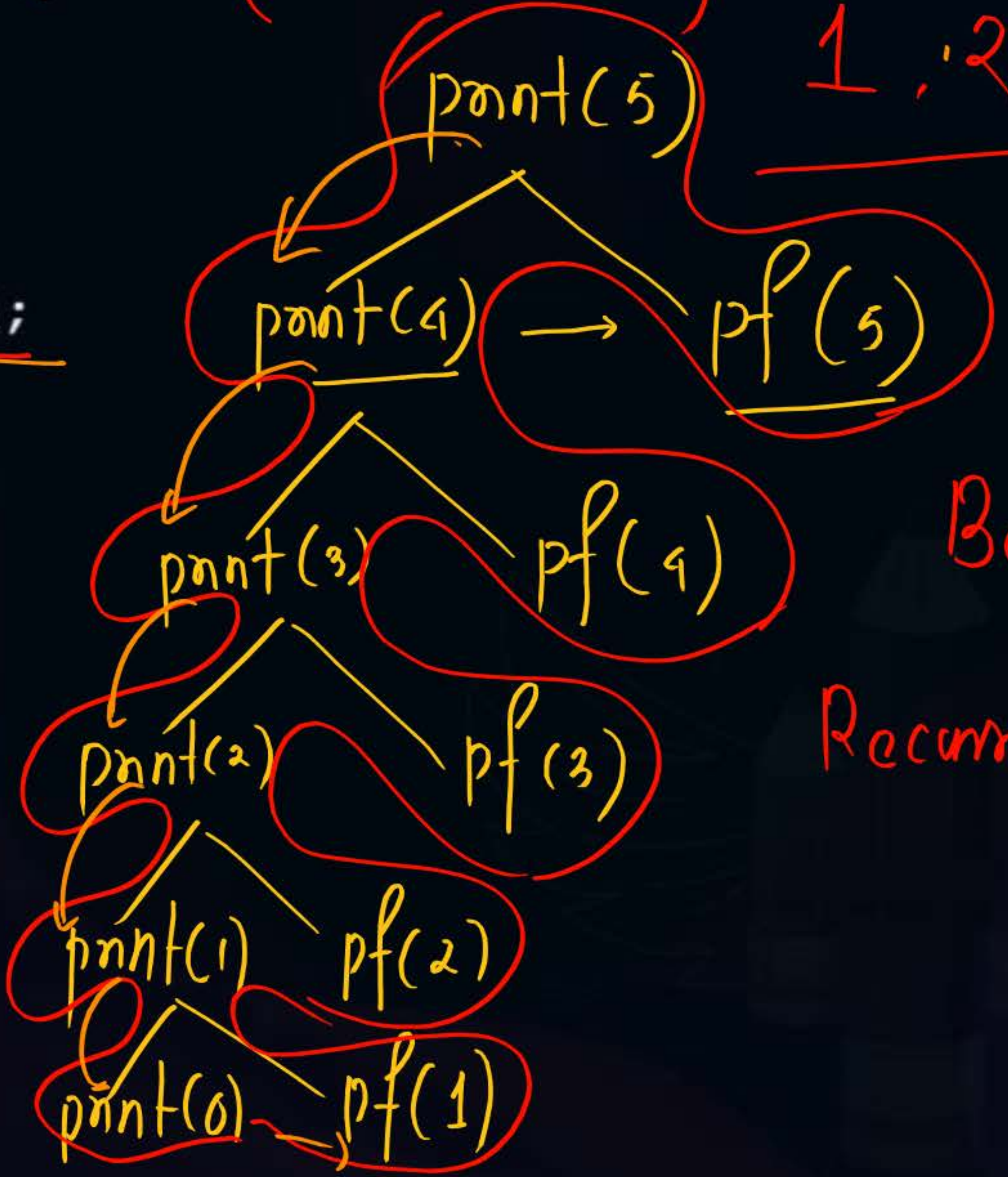
```
#include<stdio.h>
void print(int n)    {
     if (n <= 0)  return;
     print(n-1);
     printf("%d", n);
     }

int main(){
     print(5);
     return 0;
}
```

Slide

Recursion

Non Tail (Head Recursion)

```
#include<stdio.h>
void print(int n)   {
     if (n <= 0)  return;
     print(n-1);
     printf("%d", n);
     }

int main(){
     print(5);
     return 0;
}
```



1, 2, 3, 4, 5

Bottom up of

Recursion Tree

Slide

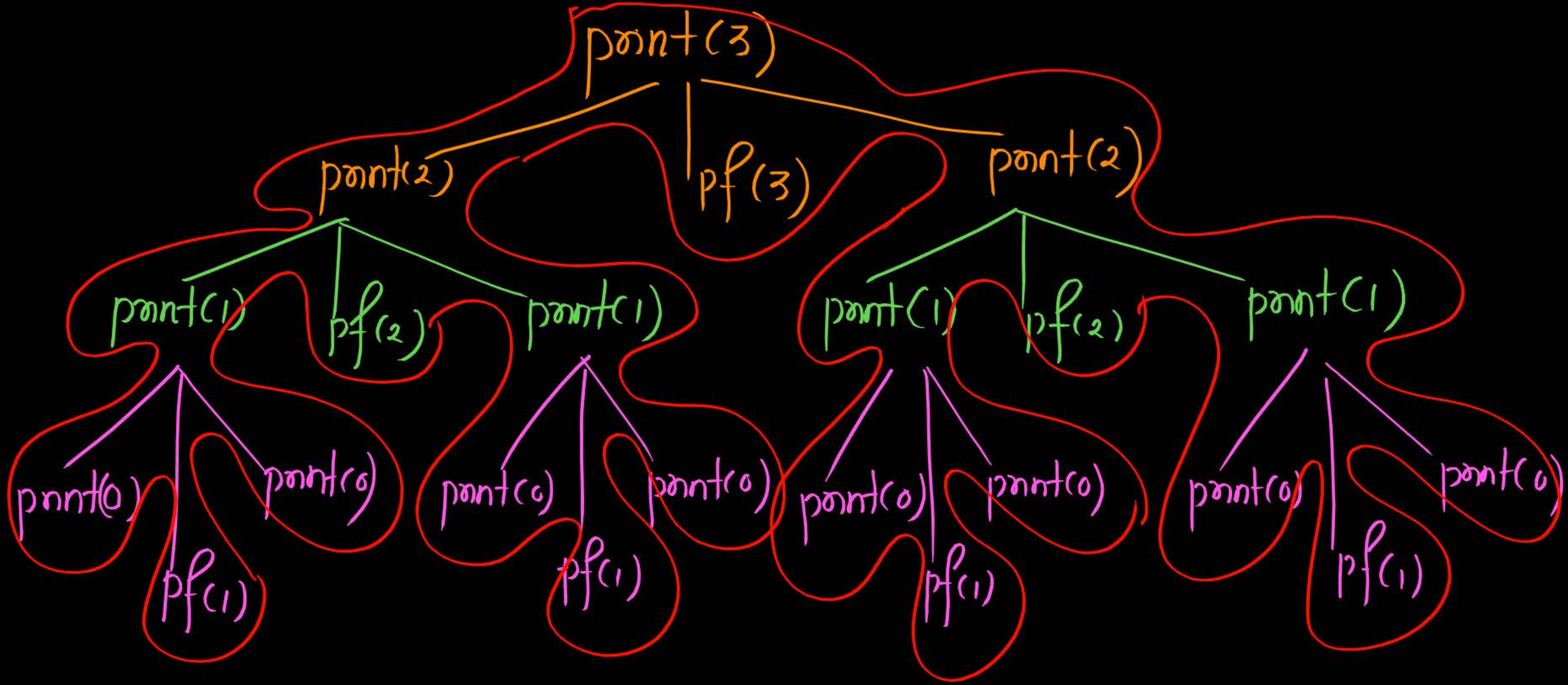
Recursion

No. of values printed. 7

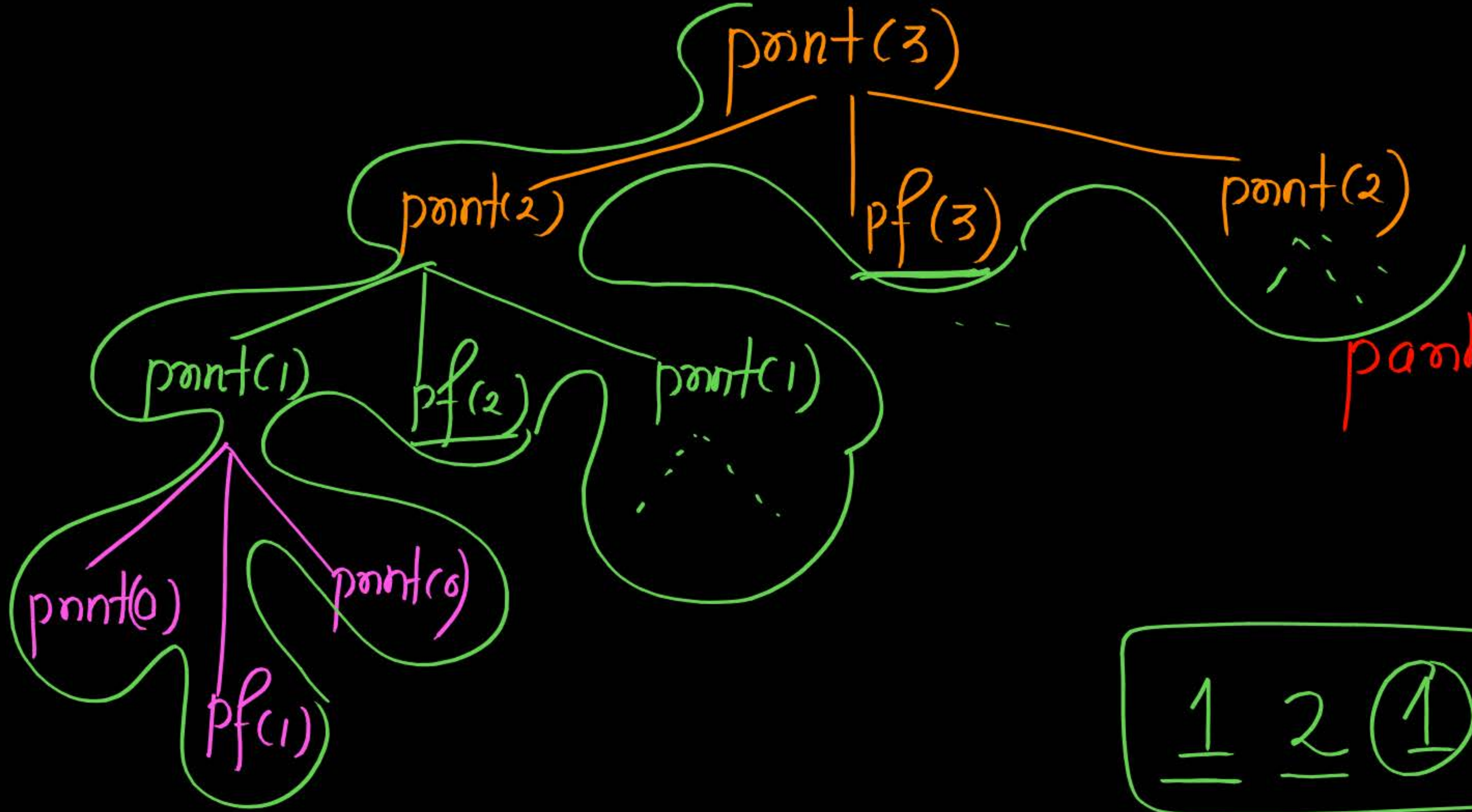
Recursion tree draw

```
#include<stdio.h>
void print(int n)    {
     if (n <= 0)  return; ✓
     print(n-1);
          printf("%d", n);
          print(n-1);
     }
int main(){
          print(3);
          return 0;
}
```

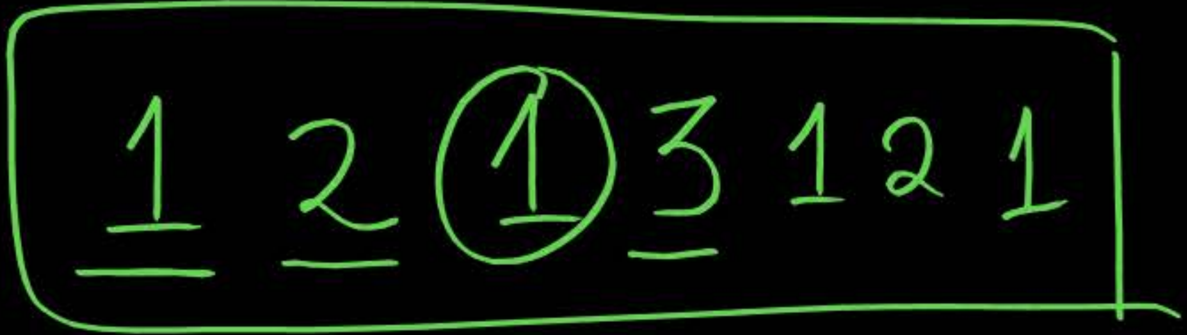
Slide


pnnt(3)
pnnt(2)
pf(3)
pnnt(2)
pnnt(1)
pf(2)
pnnt(1)
pnnt(1)
pf(2)
pnnt(1)
pnnt(0)
pf(1)
pnnt(0)
pnnt(0)
pf(1)
pnnt(0)
pnnt(0)
pf(1)
pnnt(0)
pnnt(0)
pf(1)
pnnt(0)

1 2 1 3 1 2 1 ✓



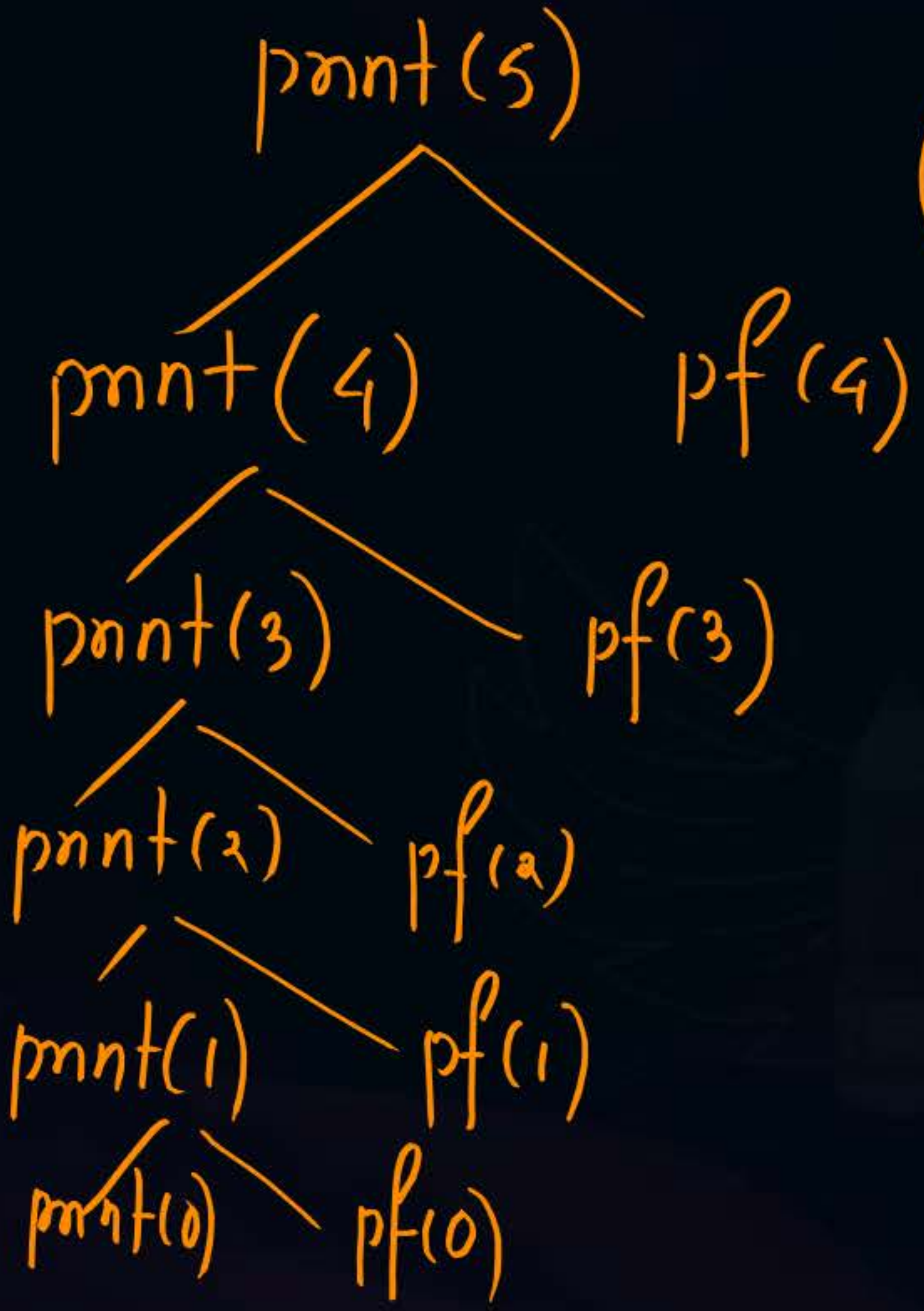
partial Recursion Tree



Recursion

```c
#include<stdio.h>
void print(int n)    {
      if (n <= 0)  return;
      print(--n);
      printf("%d", n);
      }

int main(){
      print(5);
      return 0;
}
```

No. of values printed ______



0, 1, 2, 3, 4

Slide

Recursion

```
#include<stdio.h>
void print(int n)    {
     if (n <= 0)  return;
     print(n--);
     printf("%d", n);
     }

int main(){
     print(5);
     return 0;
}
```

which of the following is true

✗ (A) 5 values pointed

✗ (B) Infinite Loop

[✓ (C) Abnormal Termination]

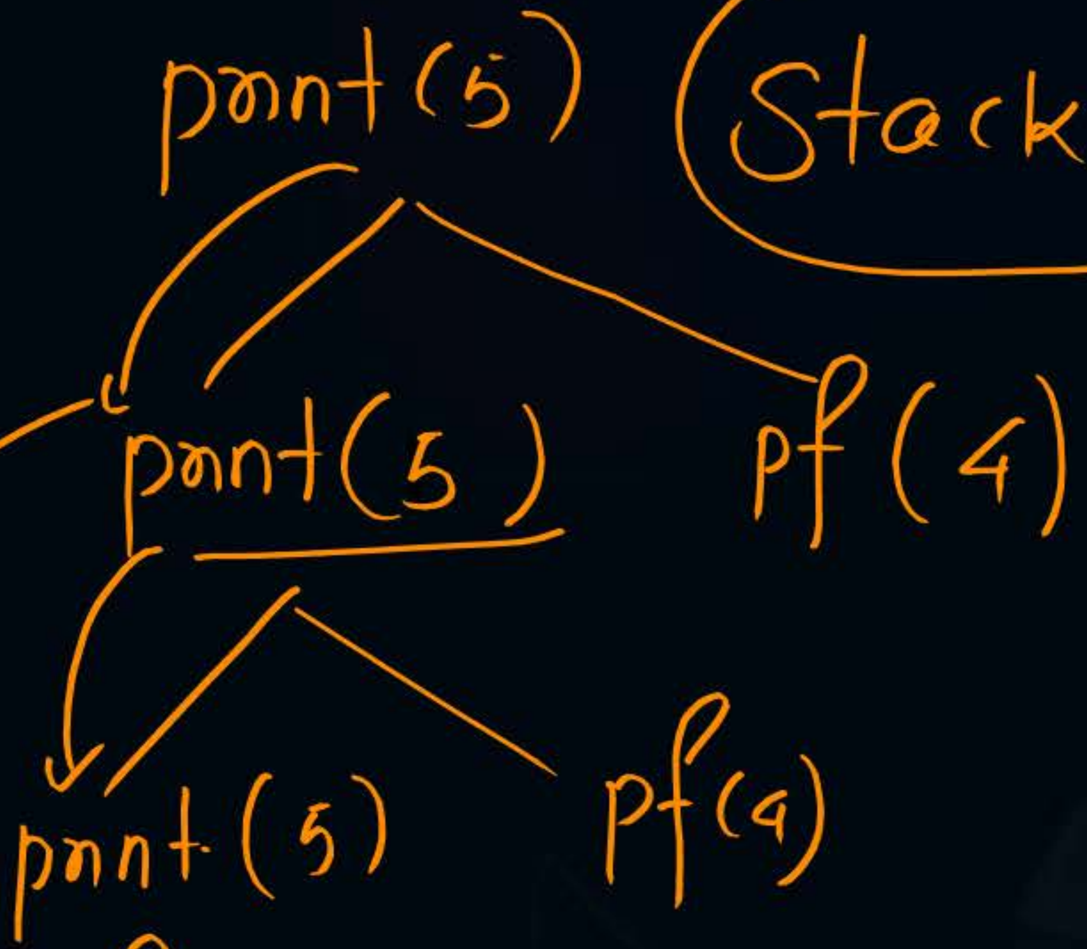
✗ (D) 4 values pointed

Slide

Recursion

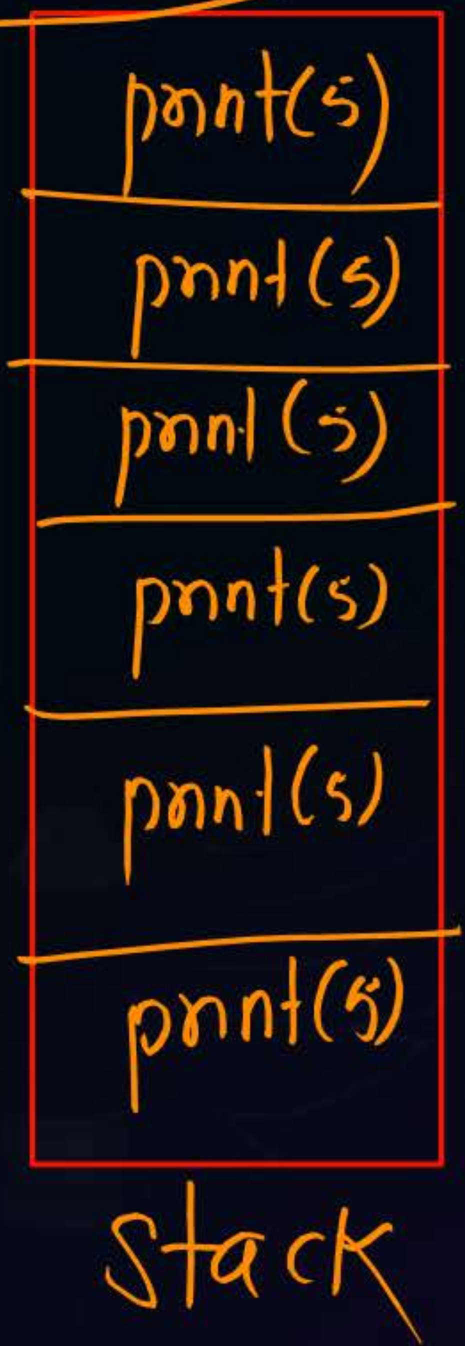
```
#include<stdio.h>
void print(int n)    {
     if (n <= 0)  return;
     print(n--);
     printf("%d", n);
     }

int main(){
     print(5);
     return 0;
}
```

(--n)



Segmentation fault !

Stack overflow



Slide

Recursion

#Q. Consider the following program

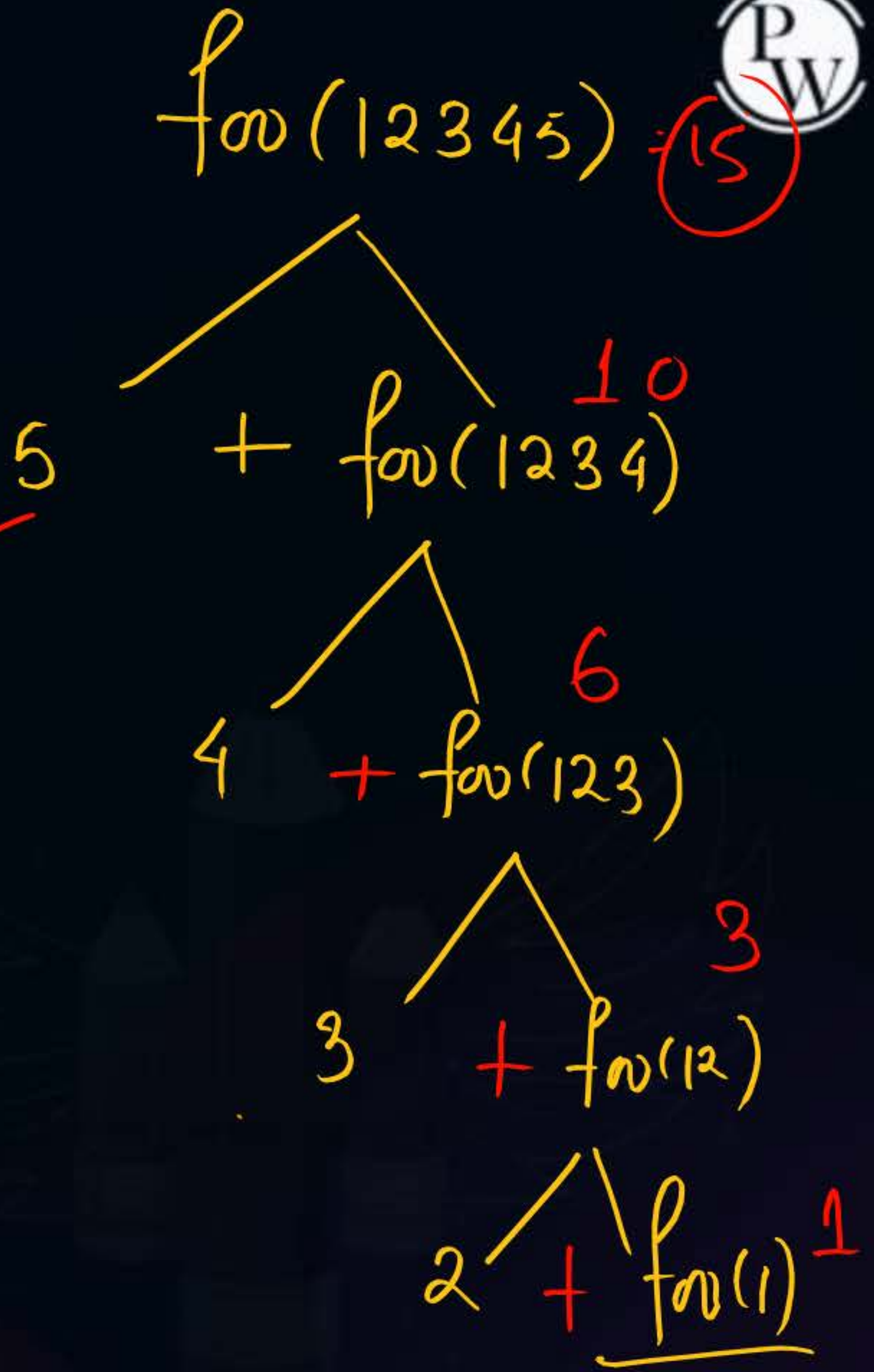
```c
#include<stdio.h>
int foo(int n){

        if (n<=9)
            return n;
    else
            return n%10+foo(n/10);
}

int main(){
        printf("%d", foo(12345));
        return 0;
}
```

A. 10

B. 11

C. 12

[✓ D. 15]



Slide

2 mins Summary

Topic Recursion

Topic Recursion Tree

Topic Tail & Nontail Recursion

Topic

Topic

Slide

THANK - YOU