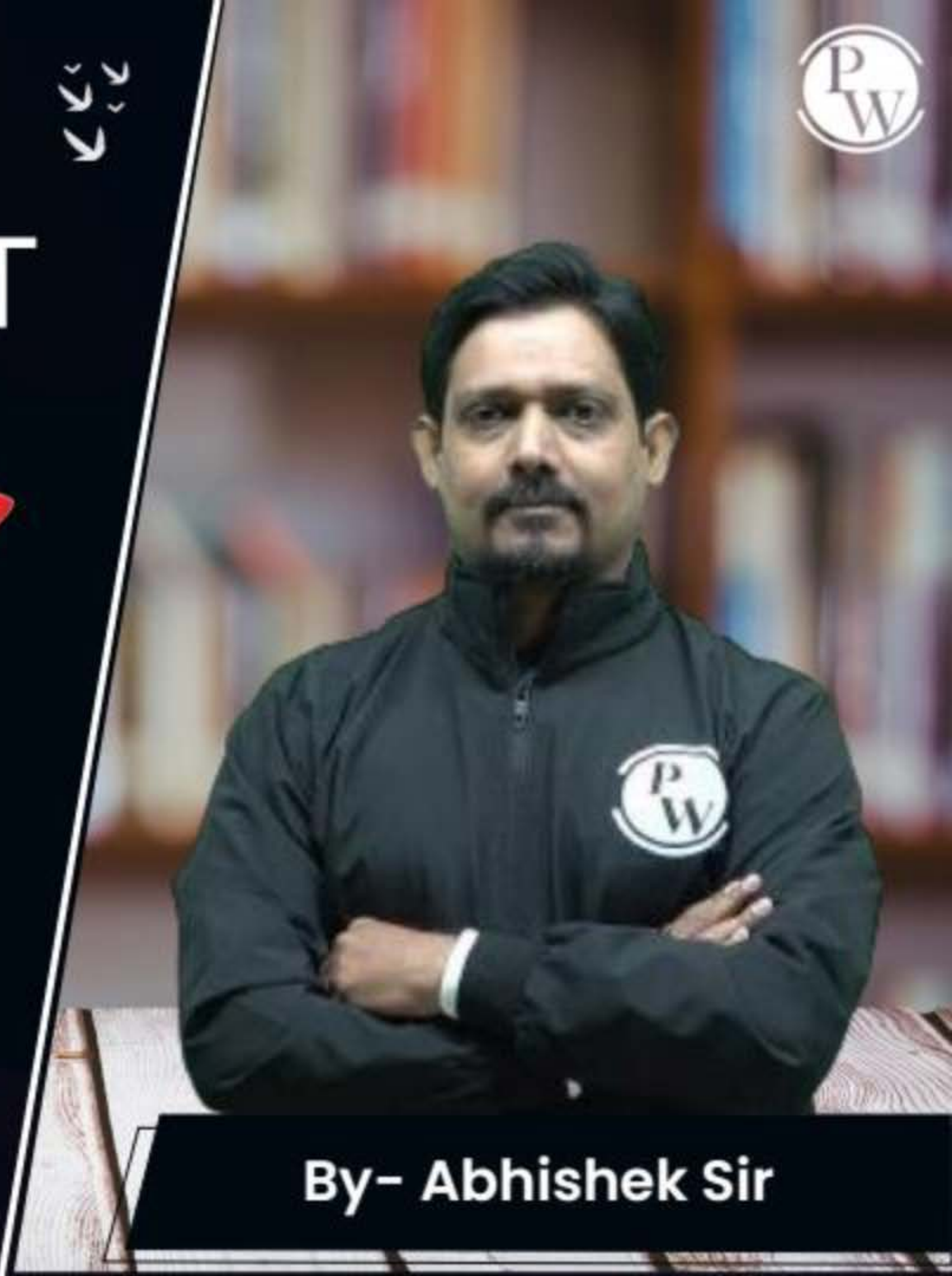
Computer Science & IT

C Programming

Function & Storage Class

Lecture No. 03



By- Abhishek Sir

Recap of Previous Lecture

Topic — Storage class

Topic — Local static — Initial value, Life-time, memory

Topic — global static — Initial value, Scope, Life time, memory

Topic — Auto.

Topic

Slide

Topics to be Covered

Topic

Recursion

Topic

Topic

Topic

Topic

Slide

1. Types of Recursion. Recursion Tree

2. output of program

3. Counting No. of function call

4 Computing value of Recursive function.

5 Establishment of Recurrence Relation from program } Algorithm

Complexity of Recursive program }

7. Solving Recurrence Relation } Discrete Mathematics

Recursion

Recursion is a problem Solving Technique in which Solution of a problem expressed as Solution of Smaller instance of same problem.

In programming Language Recursion is a function Calling itself.

Slide

Recursion

Types of Recursion

1. Tail Recursion
2. Non-Tail Recursion

} Single Recursion

3. Multiple Recursion
4. Nested Recursion
5. Indirect Recursion

Slide

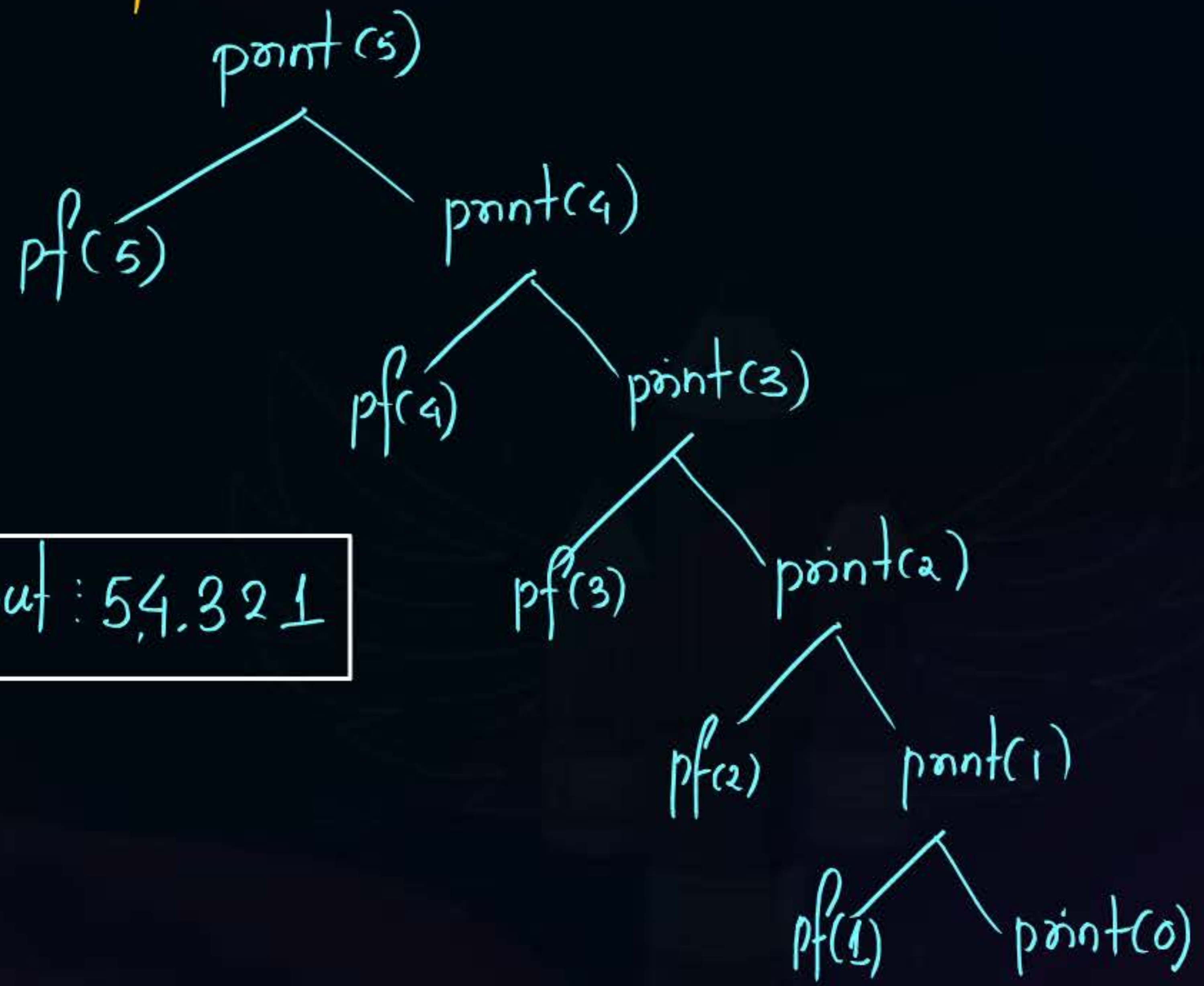
Recursion

Tail Recursion :- if Recursive call is Last statement of function then its a 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;
}
```

print(0)
print(1)
print(2)
print(3)
print(4)
print(5)

Output : 5,4.3 2 1



Slide

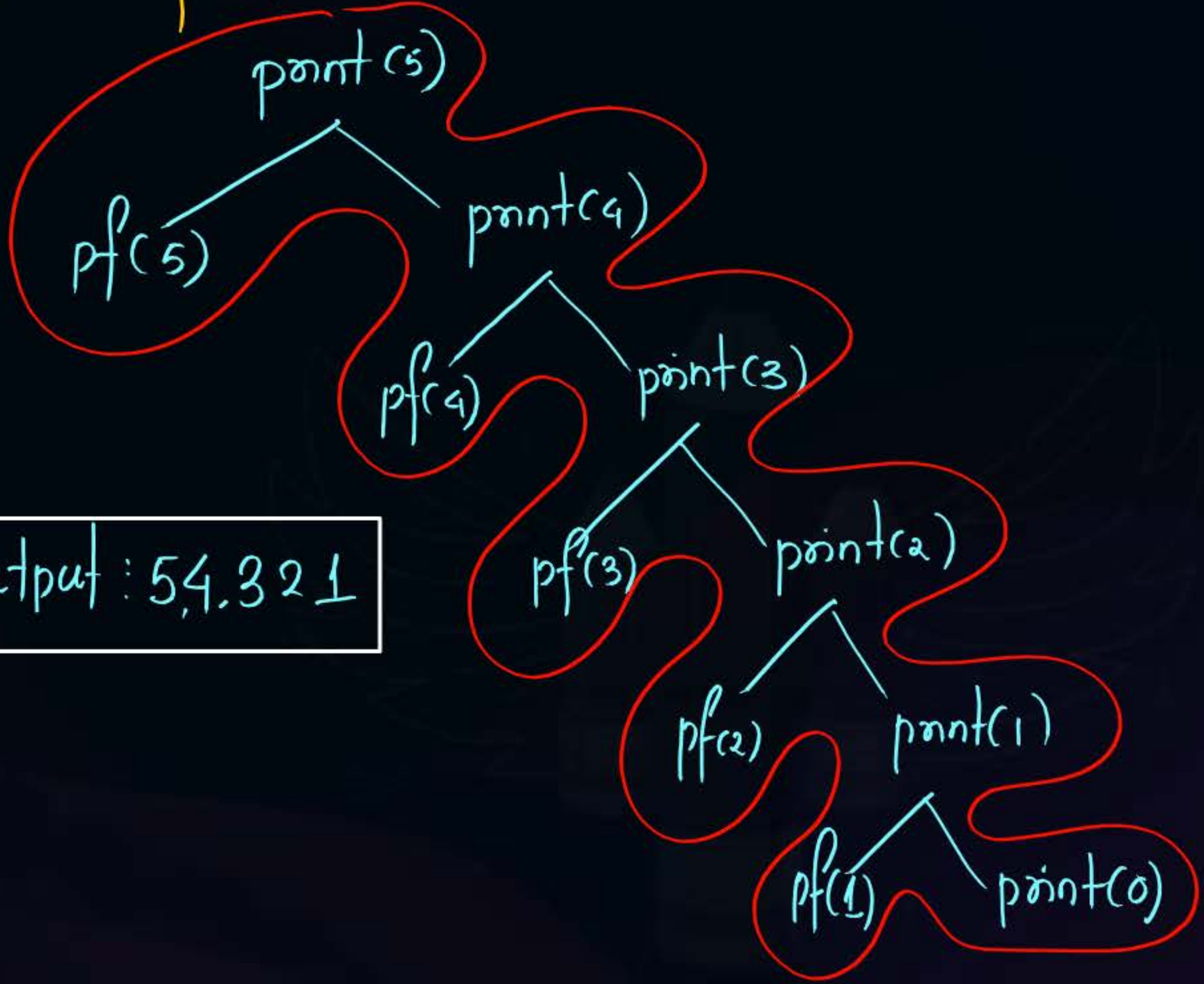
Recursion

Tail Recursion :- if Recursive call is Last statement of function then its a 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;
}
```

print(0)
print(1)
print(2)
print(3)
print(4)
print (5)

Output : 5 4 3 2 1



Slide

printf left side of Recursion tree then

value print will be in top down fashion

Recursion

Non Tail Recursion

if Recursive call is Not the Last Statement of function then we call it as Non Tail Recursion.

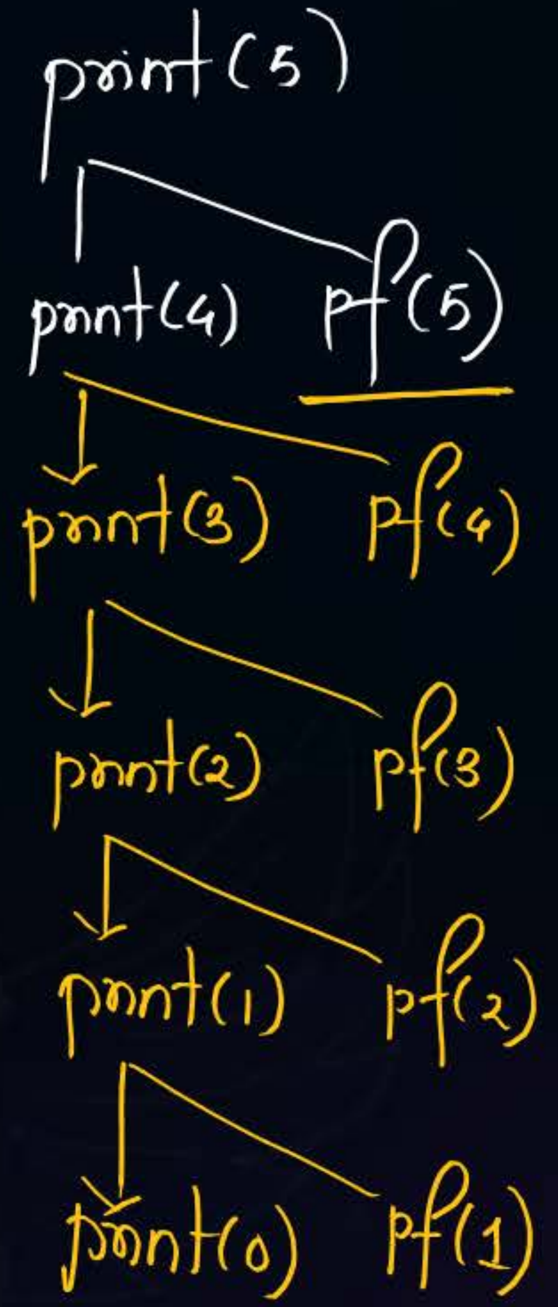
```
#include<stdio.h>
void print(int n)    {
     if (n <= 0)  return;
     print(n-1);
     printf(“%d”, n);
     }

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

Is it a modifier of value of n.??

output : 1,2,3,4,5

if printf statement Right side of Recursion tree then it will be printed in Bottom up manner.



Recursion

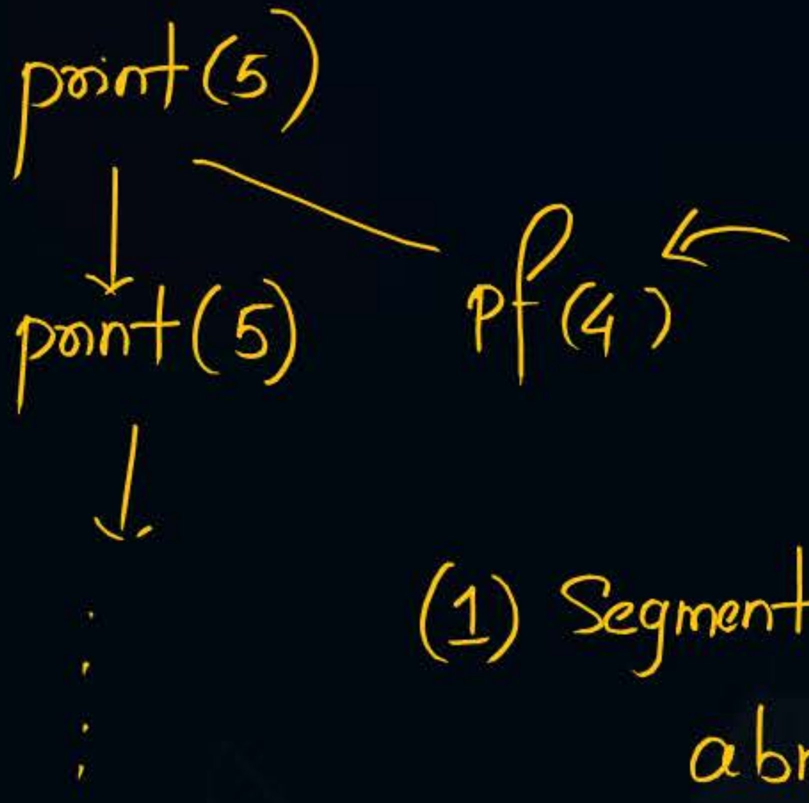
```
#include<stdio.h>
void print(int n)    {
        if (n <= 0)  return;
        print(n--);
        printf(“%d”, n);
        }

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

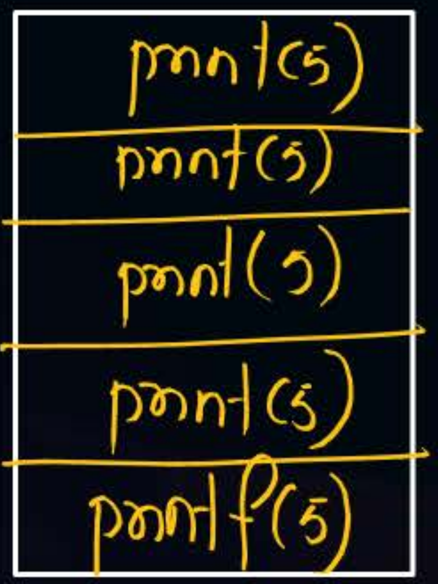
postdecrement

use old value

exhaust Run time Stack.



this statement



(1) Segmentation fault

abnormal Termination

(3) infinite Loop

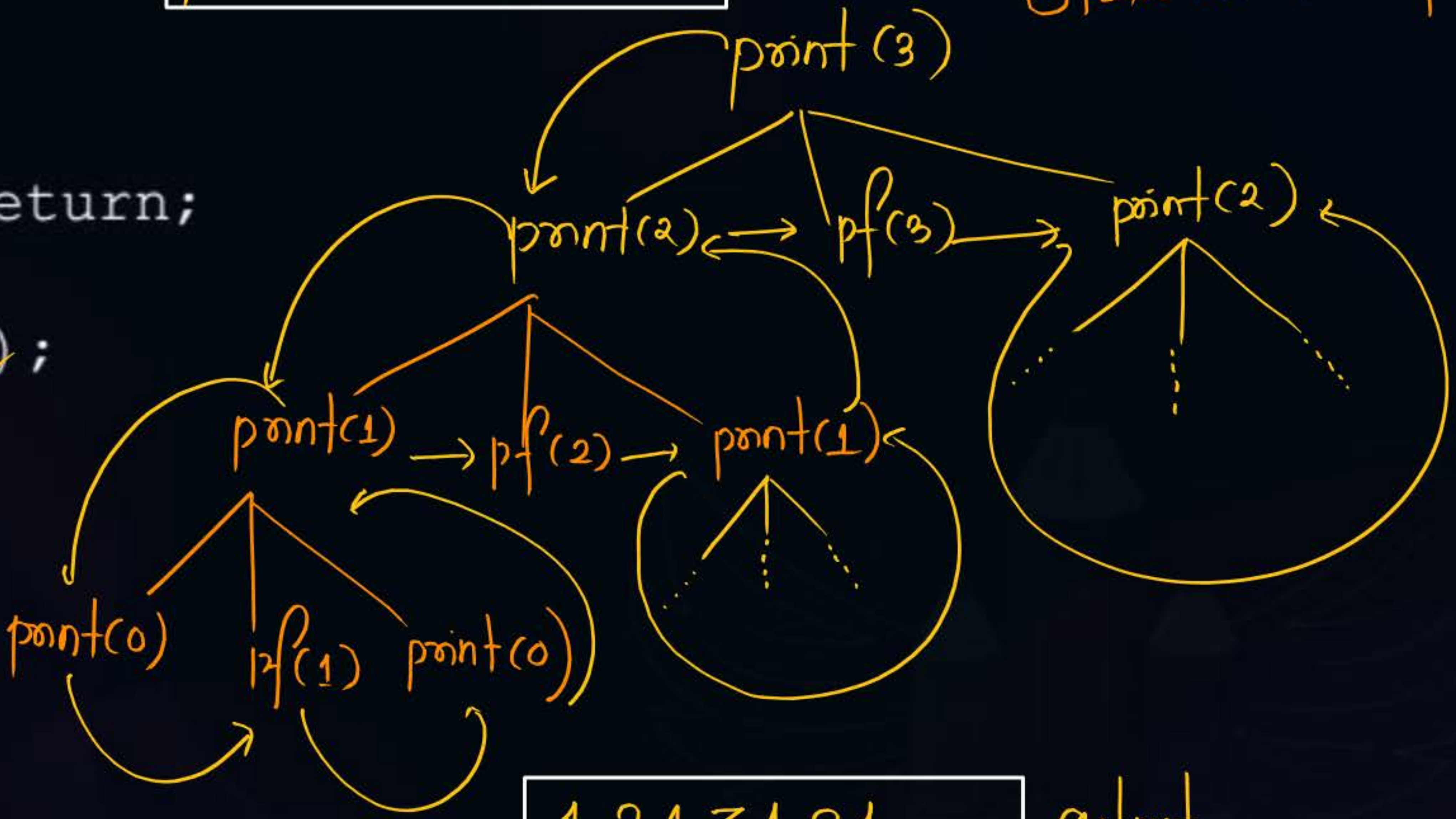
Slide

Recursion

```
#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;
}
```

partial Recursion Tree

Dont use if
Static variable is present



1 2 1 3 1 2 1

output.

Slide

Recursion

#Q. Consider the following program

```
#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

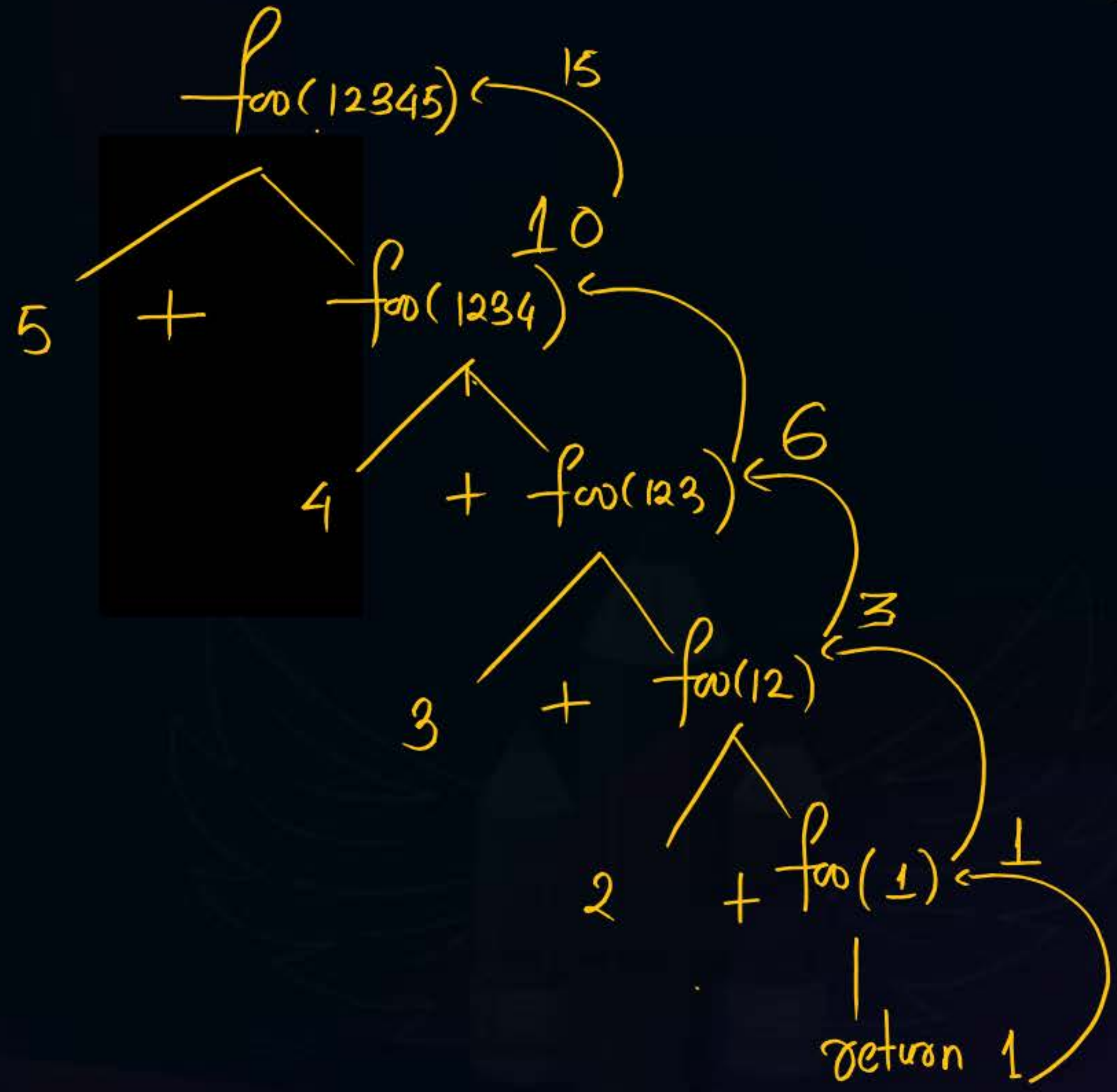
Recursion

#Q. Consider the following program

```
#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;
}
```



Slide

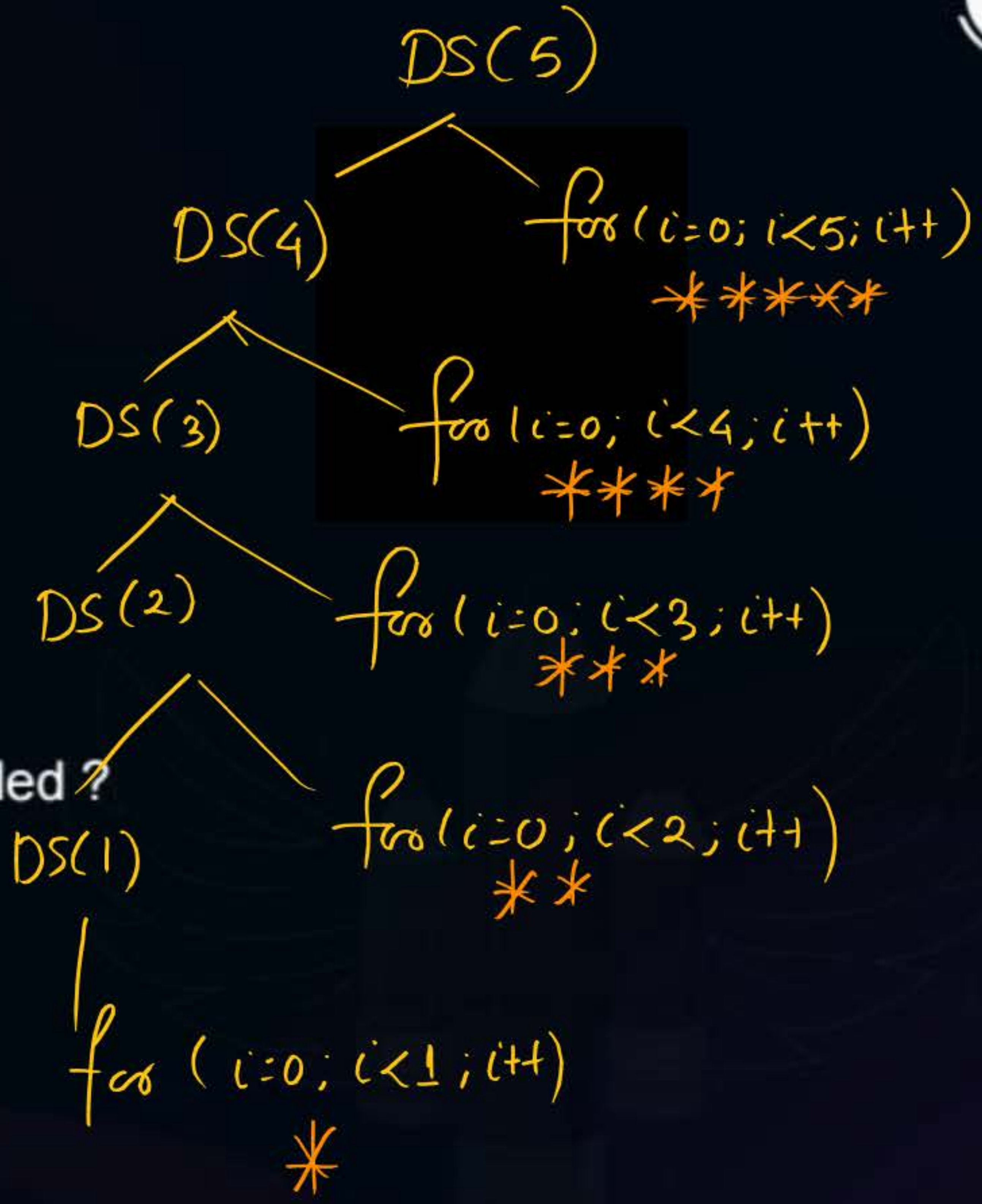
Recursion

#Q. Consider the following program

```
void Dosomething(int n) {
      if (n > 1)
            Dosomething(n-1);
      for (int i = 0; i < n; i++)
            printf("*");
      printf("\n");
}
```

The number of stars will print if the Dosomething(5) is called ?

[15]



Recursion

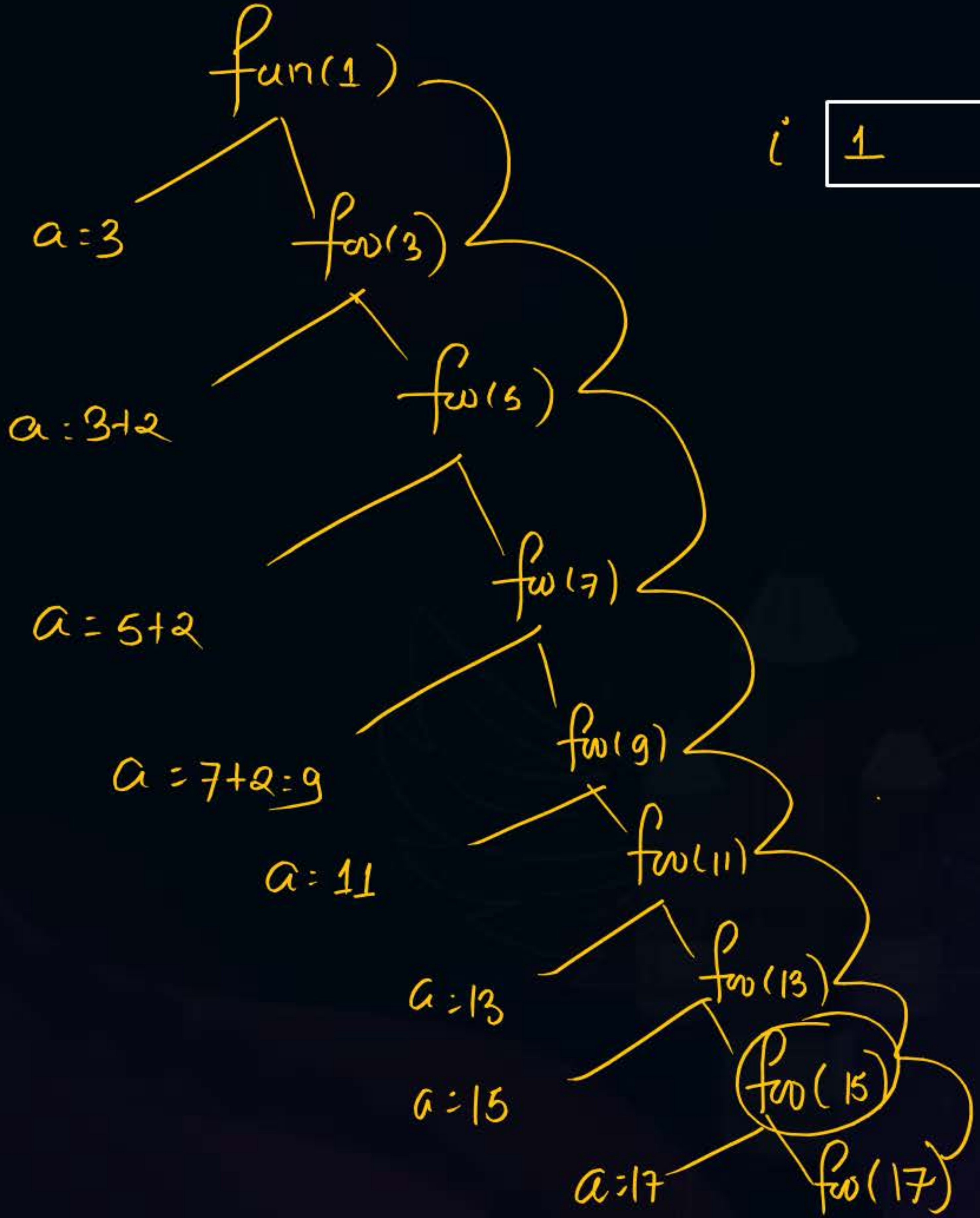
Consider the following C code

```
int fun(int a){
static int i;
   i = 1;
   if (a >15) return a;
   a = a + 2*i;
   i++;
   return fun(a);
}
```

Single copy

15>15 false

The value returned by fun(1) is ______



i 1

Slide

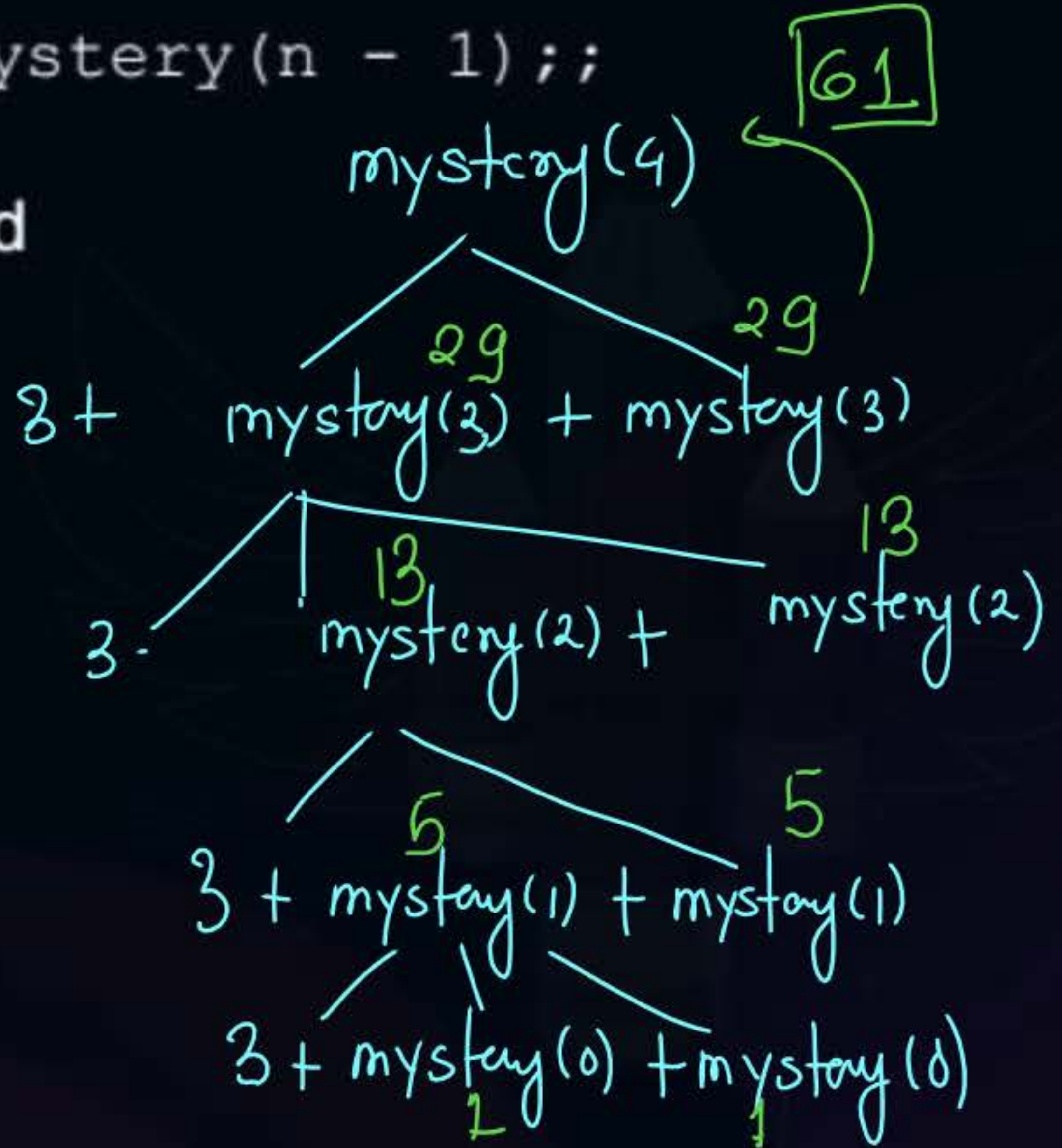
Question

#Q. Consider the following program

```
int mystery(int n) {
        if (n <= 0)
                return 1;
        else
                return 3 + mystery(n - 1) + mystery(n - 1);;
}
```

What is the output the program when mystery(4) is called

(A) 59
(B) 50
[✓ (C) 61]
(D) 62



61
mystery(4)
29
29
3+ mystory(3) + mystery(3)
13
13
3 mystery(2) + mystery(2)
5
5
3 + mystory(1) + mystory(1)
3 + mystory(0) + mystory(0)
1
1

Slide

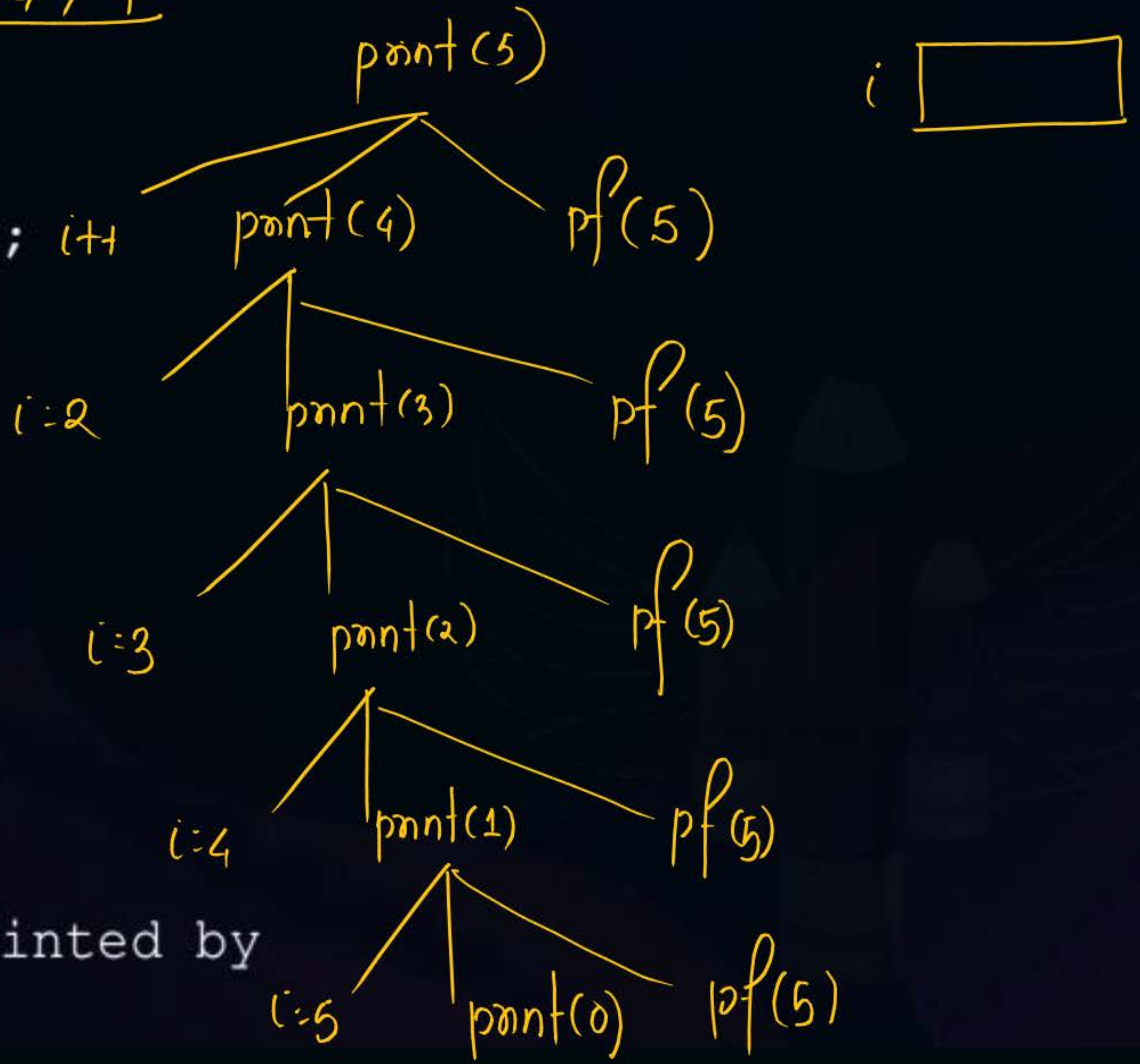
5,5,5,5,3

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

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

The sum of the value printed by the program is

Slide



Question

```c
#include<stdio.h>
int mystery(int n, int a, int r){
        if (n==1)
                return a;
        else
                return a + r*mystery(n-1, a, r);
}
}
```

What will be returned by the called mystery(4,10, 2)?

[redacted]

(A) 200

(B) 140

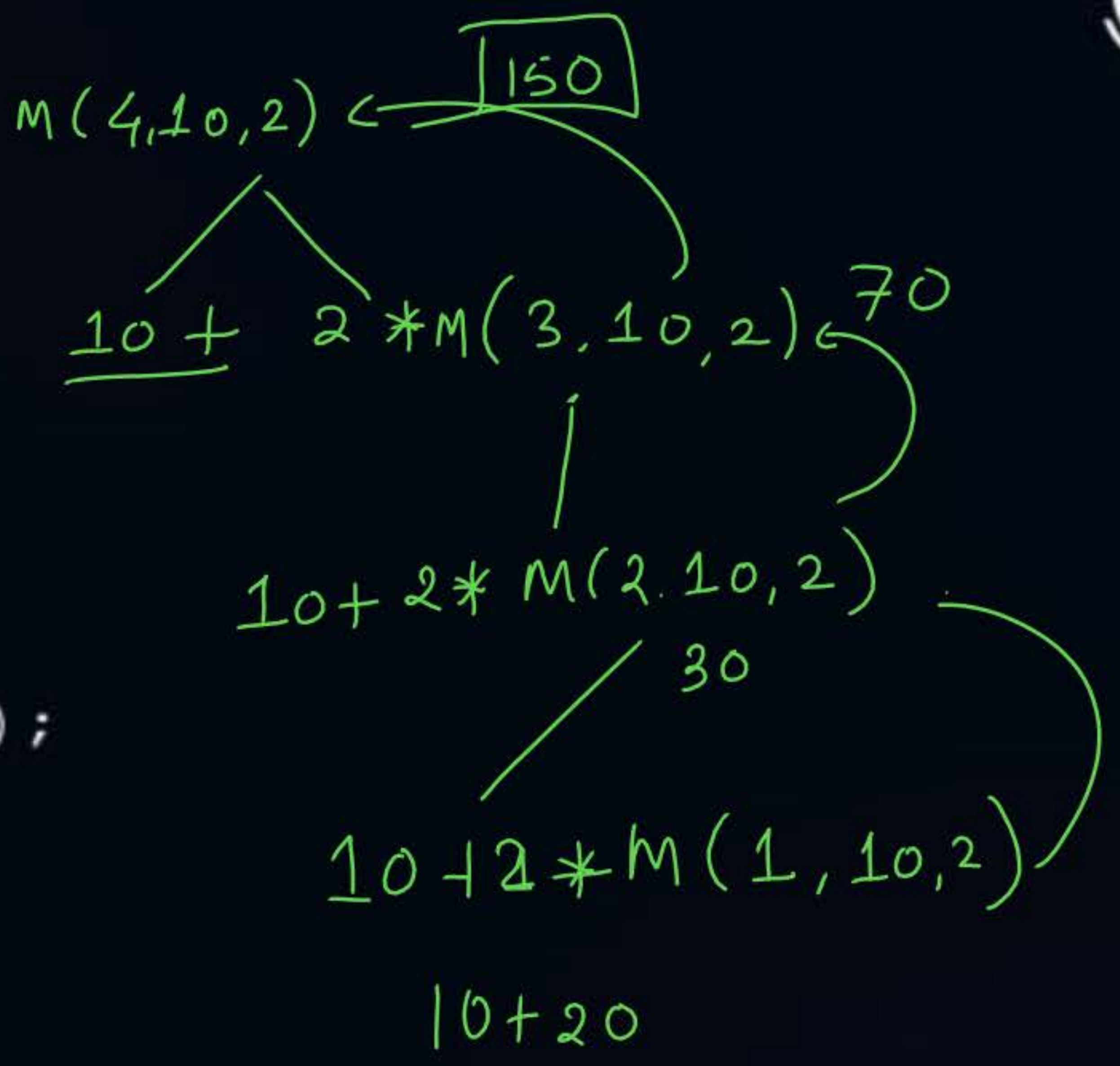
(C) 100

(D) None of the above

Question

```
#include<stdio.h>
int mystery(int n, int a, int r){
        if (n==1)
                return a;
        else
                return a + r*mystery(n-1, a, r);
}
}
```

What will be returned by the called mystery(4,10, 2)?



mystary (10, 10,2)

Question

```c
#include<stdio.h>
int mystery(int n, int a, int r){
        if (n==1)
                return a;
        else
                return a + r*mystery(n-1, a, r);
}
}
```

What will be returned by the called mystery(4,10, 2)?

mystary (10, 10,2)

$a+ar+ar^2+ar^3$

mystery$(4, a, r)$

$a+r(a+ar+ar^2)$

$a + r * M(3, a, r)$ $= a+ar+ar^2 + ar^3$

$a+r(a+ar)$

$a + r * M(2, a, r)$ $= a+ar+ar^2$

$a + r * M(1, a, r)$

$$= \frac{a(r^n-1)}{r-1} = \frac{10(2^{10}-1)}{2-1}$$

$$= 10 \times 1030$$

$$\boxed{10230}$$

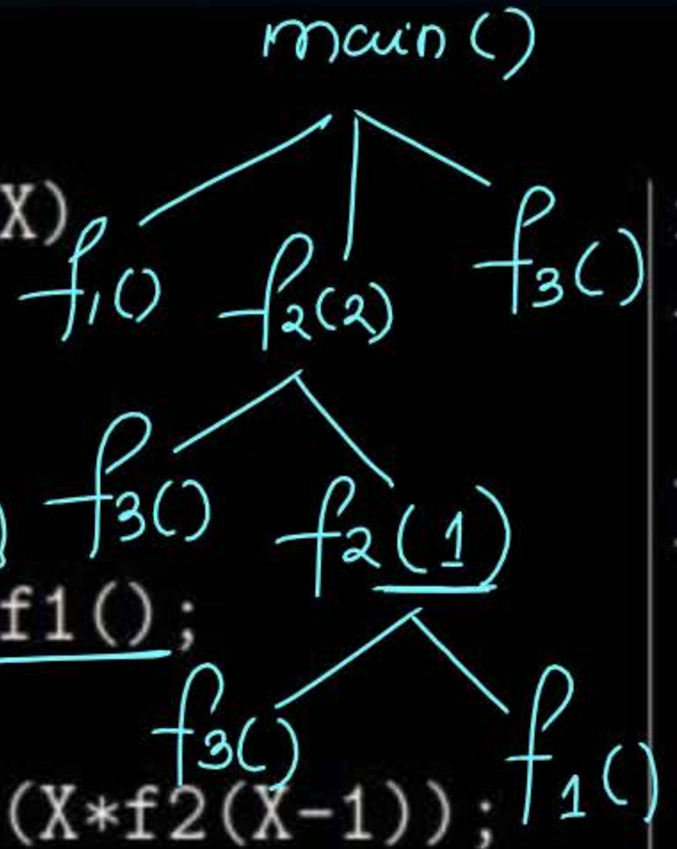
Consider the following program:

```
int main()
{
   f1();
   f2(2);
   f3();
   return(0);
}
```

```
int f1()
{
  return(1);
}
```

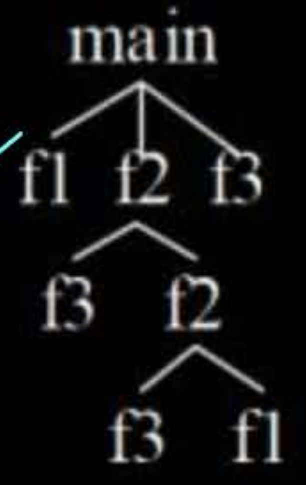
```
int f2(int X)
{
   f3();
  if (X==1)
    return f1();
  else
    return (X*f2(X-1));
}
```

```
int f3()
{
  return(5);
}
```

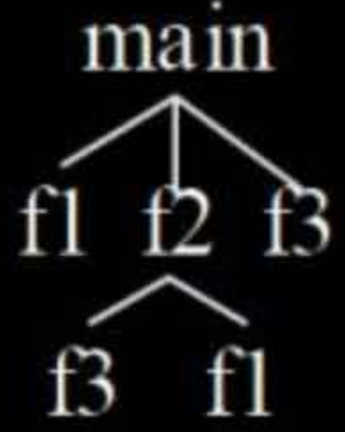


Which one of the following options represents the activation tree corresponding to the main function?

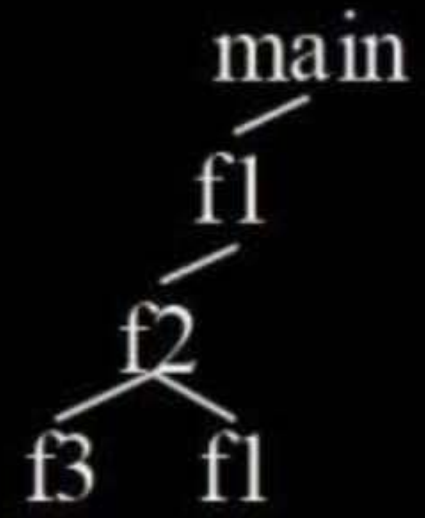
[✓ (A)]



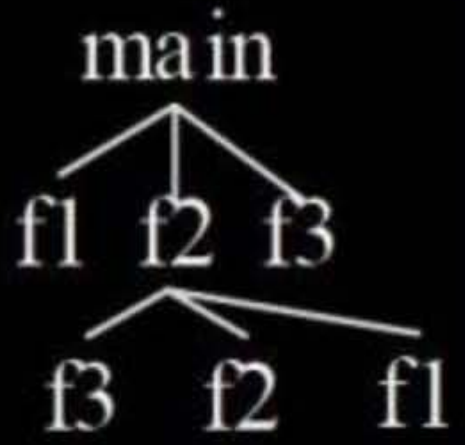
(B)



(C)



(D)



Consider the following program:

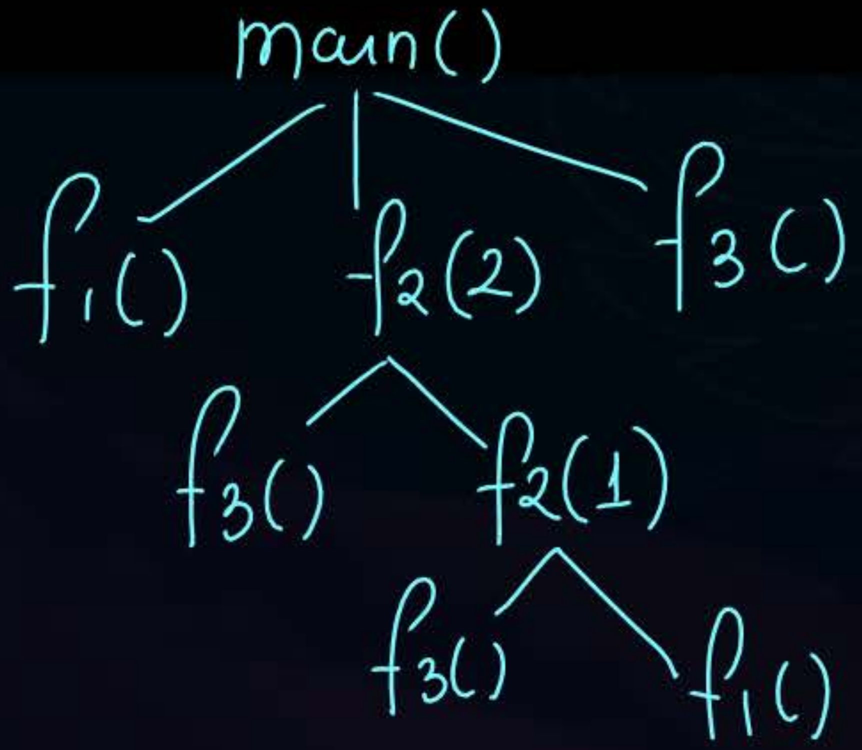
```
int main()
{
    f1();
    f2(2);
    f3();
    return(0);
}
```

```
int f1()
{
  return(1);
}
```

```
int f2(int X)
{
    f3();
  if (X==1)
    return f1();
  else
    return (X*f2(X-1));
}
```

```
int f3()
{
  return(5);
}
```

Which one of the following options represents the activation tree corresponding to the main function?



Slide

2 mins Summary

Topic

Recursion

Topic

Topic

Topic

Topic

Slide

THANK - YOU