CS & IT ENGINEERING

C-Programming

Function & Storage Class

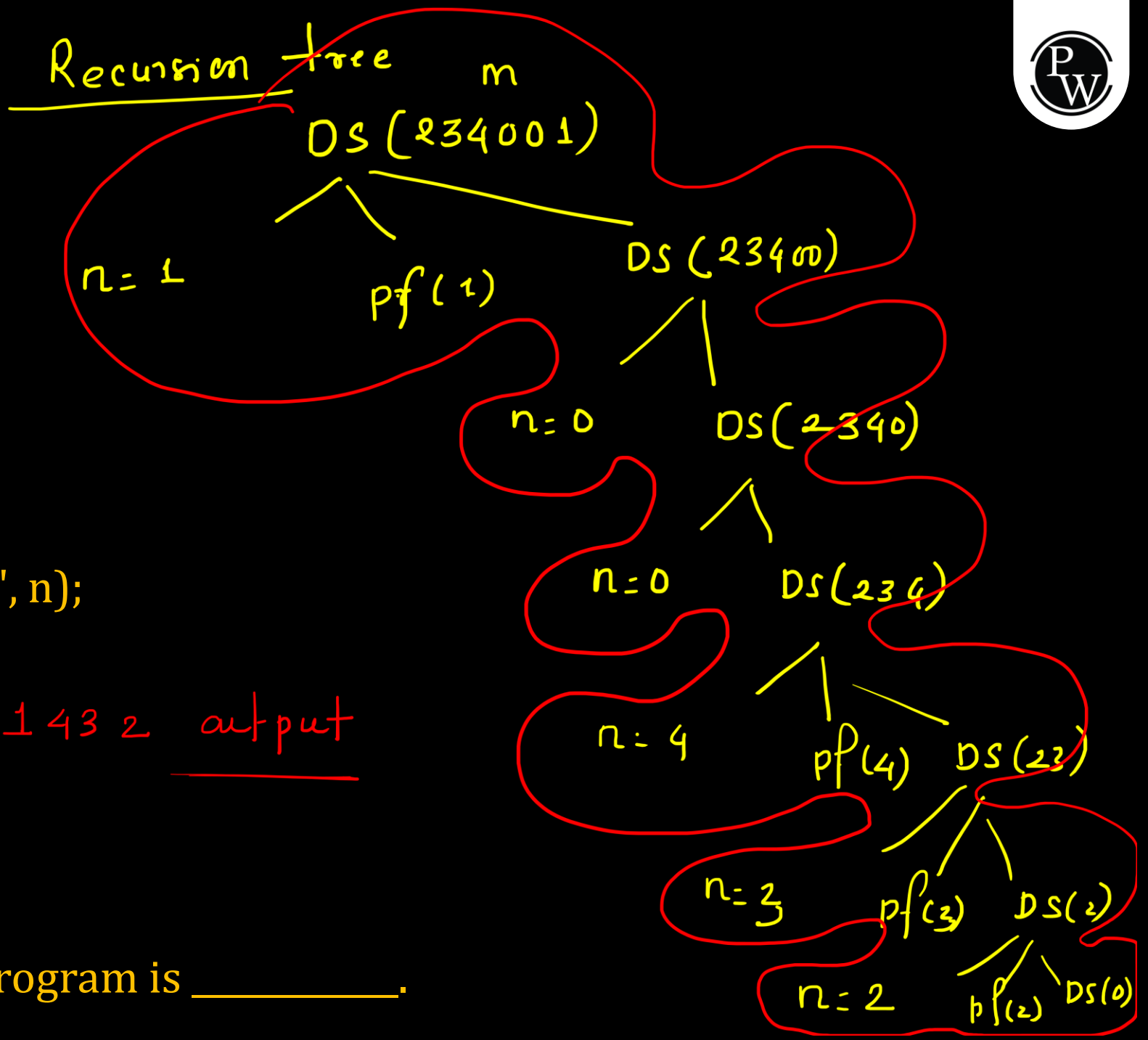
DPP 04 Discussion Notes

By- Abhishek Sir

Question

#Q.
```
#include <stdio.h>
void DS( int m){
        int n;
        if (m==0)
                return;
        n = m %10;
        if (n!=0)
                printf("%d", n);
        DS(m/10);
        return;
        }
int main() {
DS(234001);
}
```

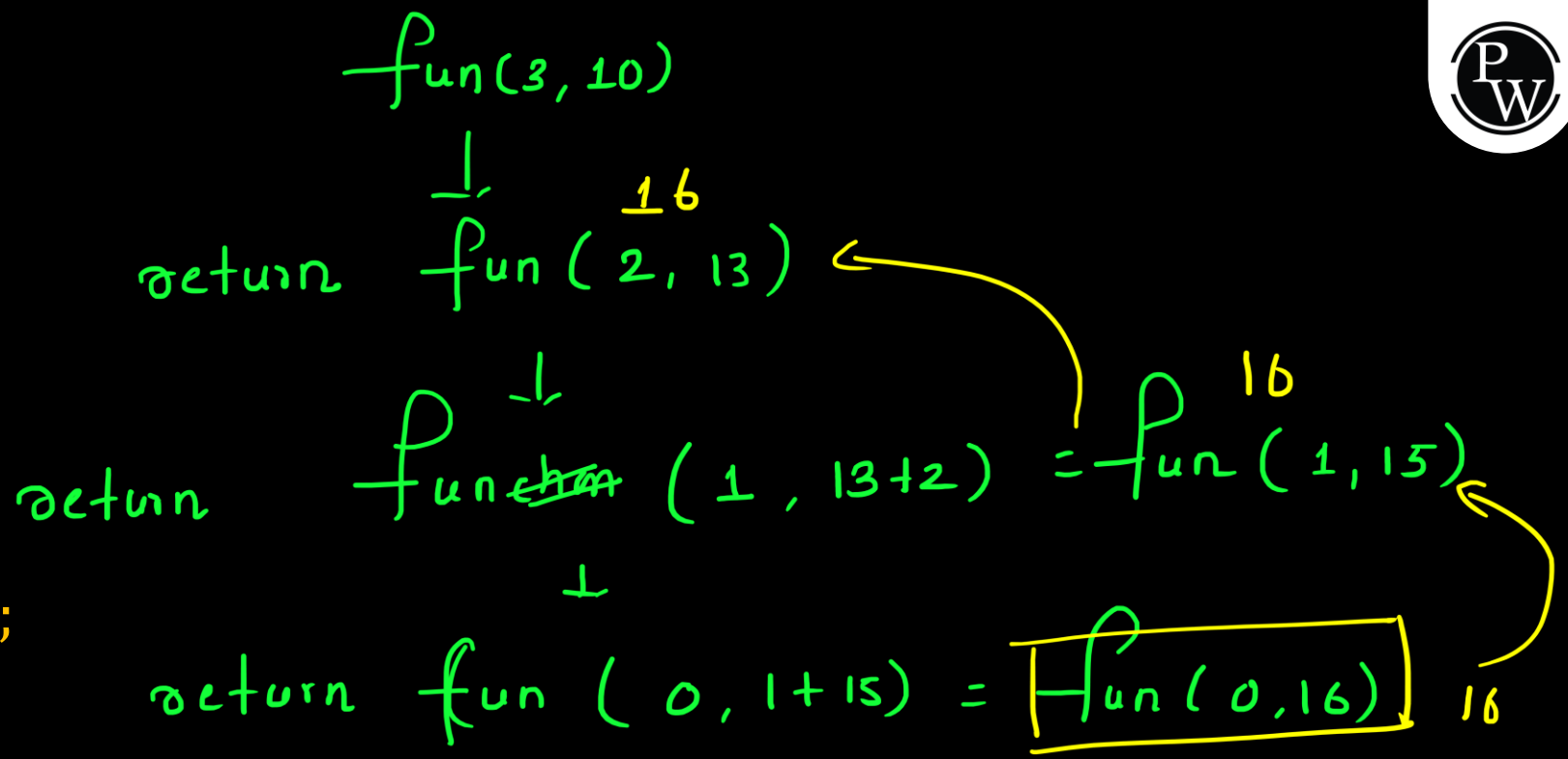
The value printed by the program is _____.



Question

#Q.
```
#include <stdio.h>
int fun(int x, int y) {
    if (x == 0)
        return y;
    else
        return fun(x - 1, x + y);
}
int main(){
   printf("%d", fun(3,10));
   return 0;
}
```

The output of the program is



[C]

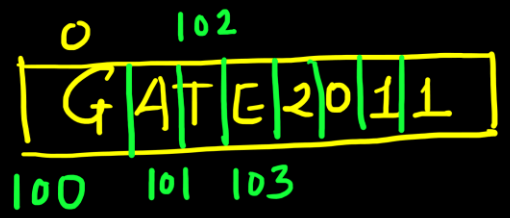
A 12

B 15

C 16

D 17

Question



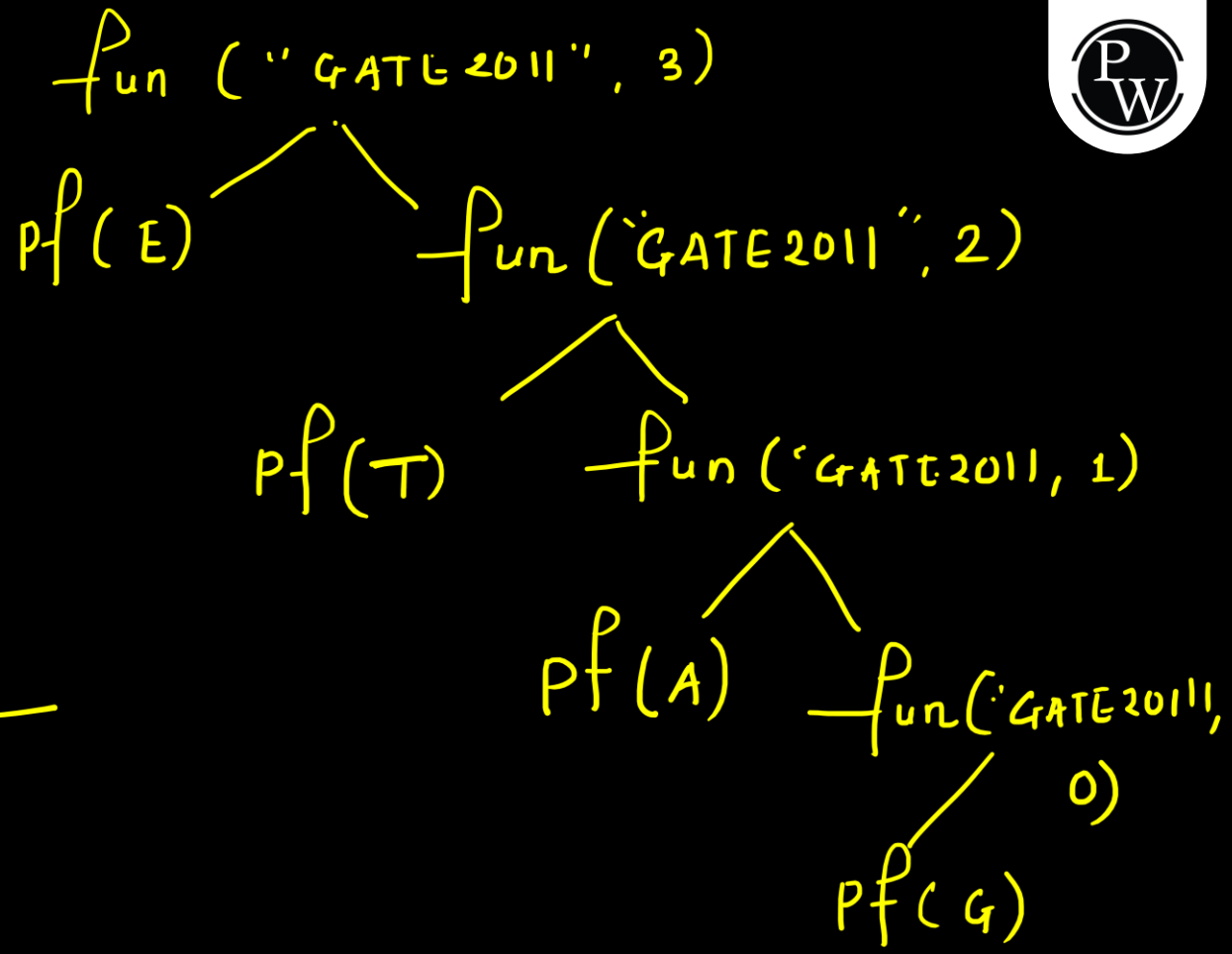
#Q. Consider the following program

```
#include <stdio.h>
void fun(char * a,int len){
    if(len==0)
        printf("%c",a[len]);
    else {
        printf("%c",a[len]);
        fun(a,len-1);
    }
}
int main(){
    fun("GATE2011", 3) ;
}
```

The output the program us

ETAG

[C]



A EGATE

B GATE

C ETAG

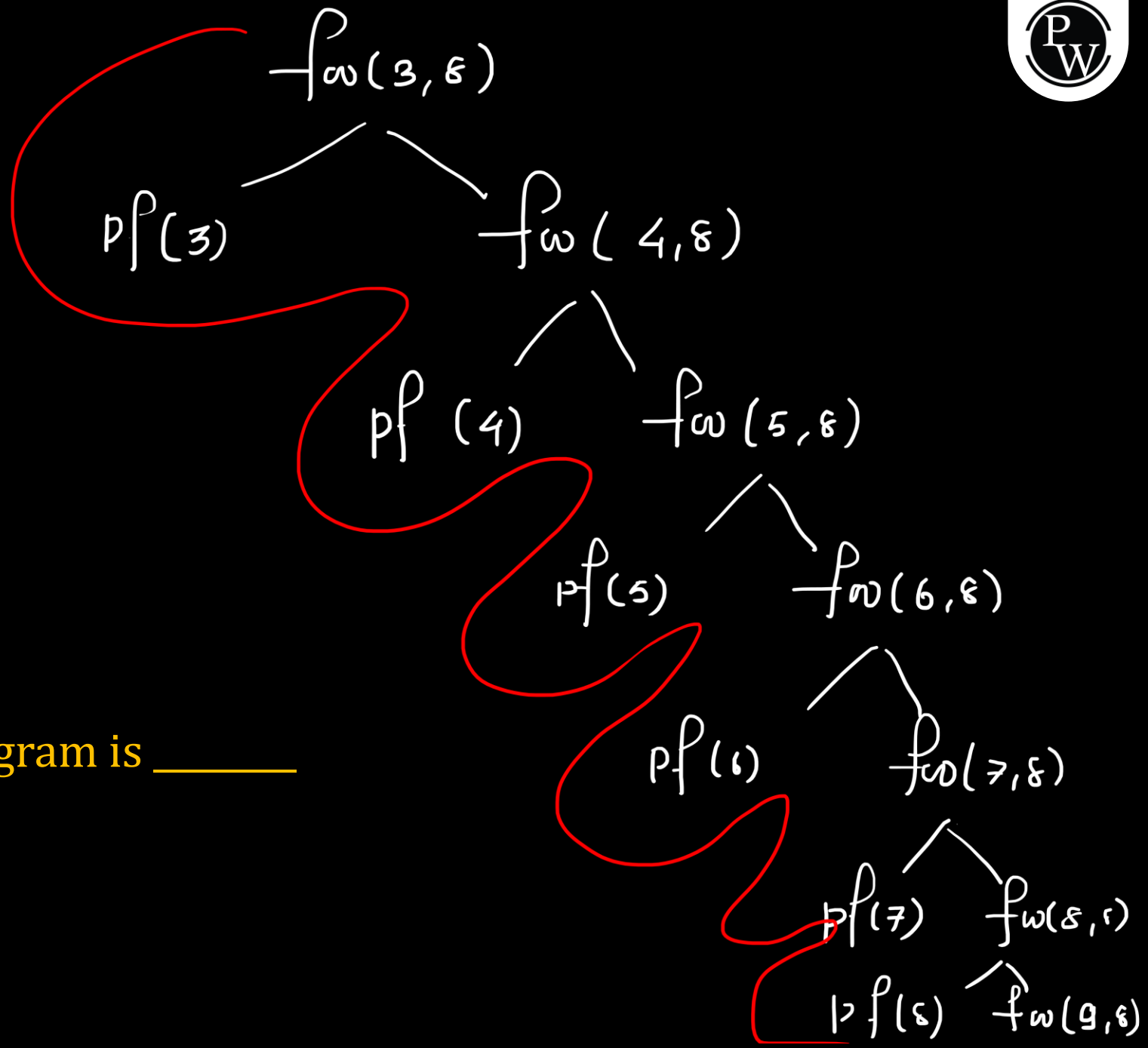
D TAGE

Question

#Q.
```
#include <stdio.h>
void foo(int left, int right) {
    if (left <= right) {
    printf("%d",left);
    foo(left + 1, right);
        }
        }
int main(){
    foo(3,8) ;
}
```

The value printed by the program is _____

3,45678



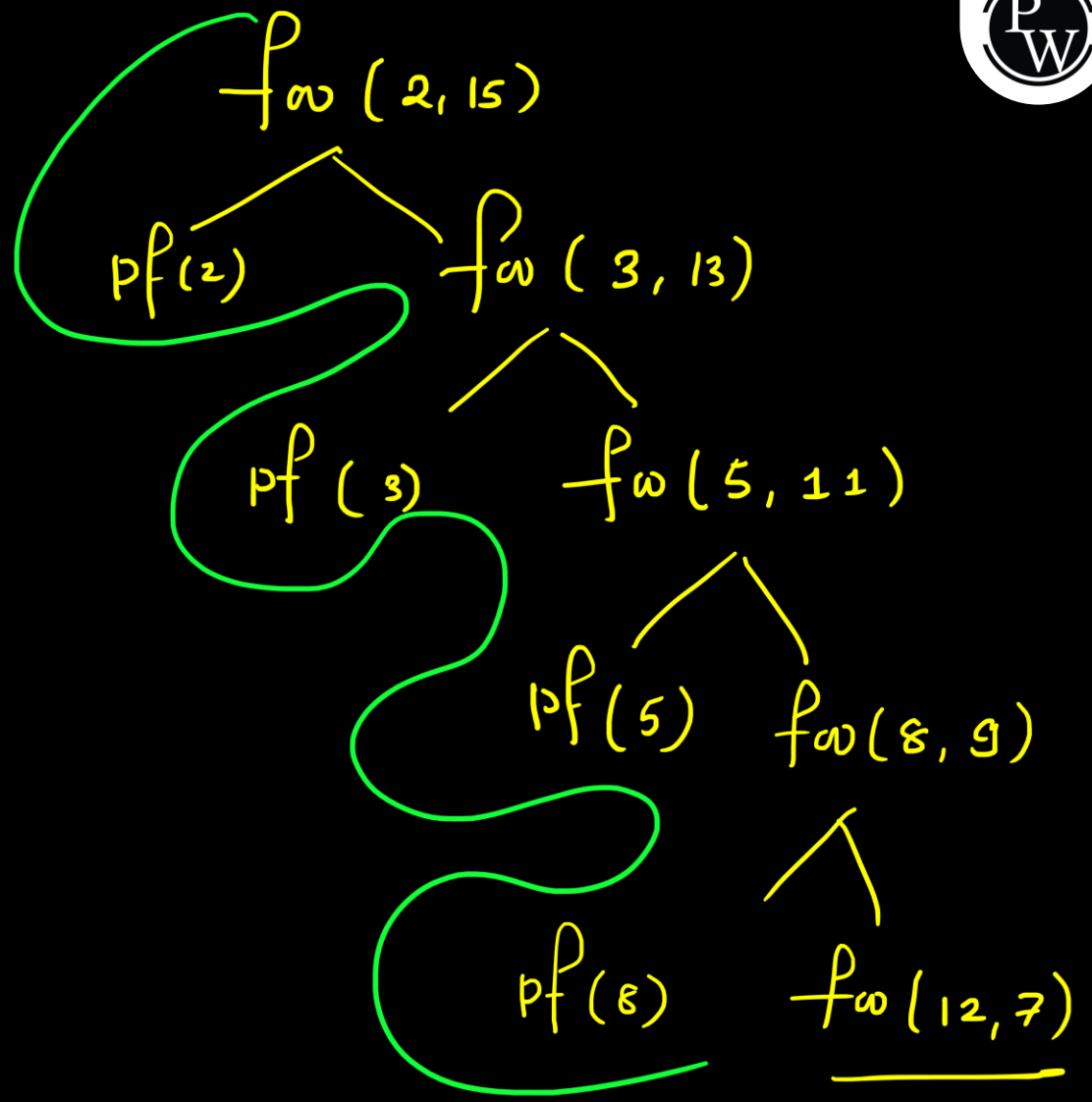
Question

Value preserve

count ⇒ 1/2/3/4

#Q.
```
#include <stdio.h>
 void foo(int left, int right) {
    static count;
    if (left <= right) {
    count++;
    printf("%d",left);
    foo(left+count, right-2);
      }
    }

 int main(){
   foo(2, 15) ;
 }
```



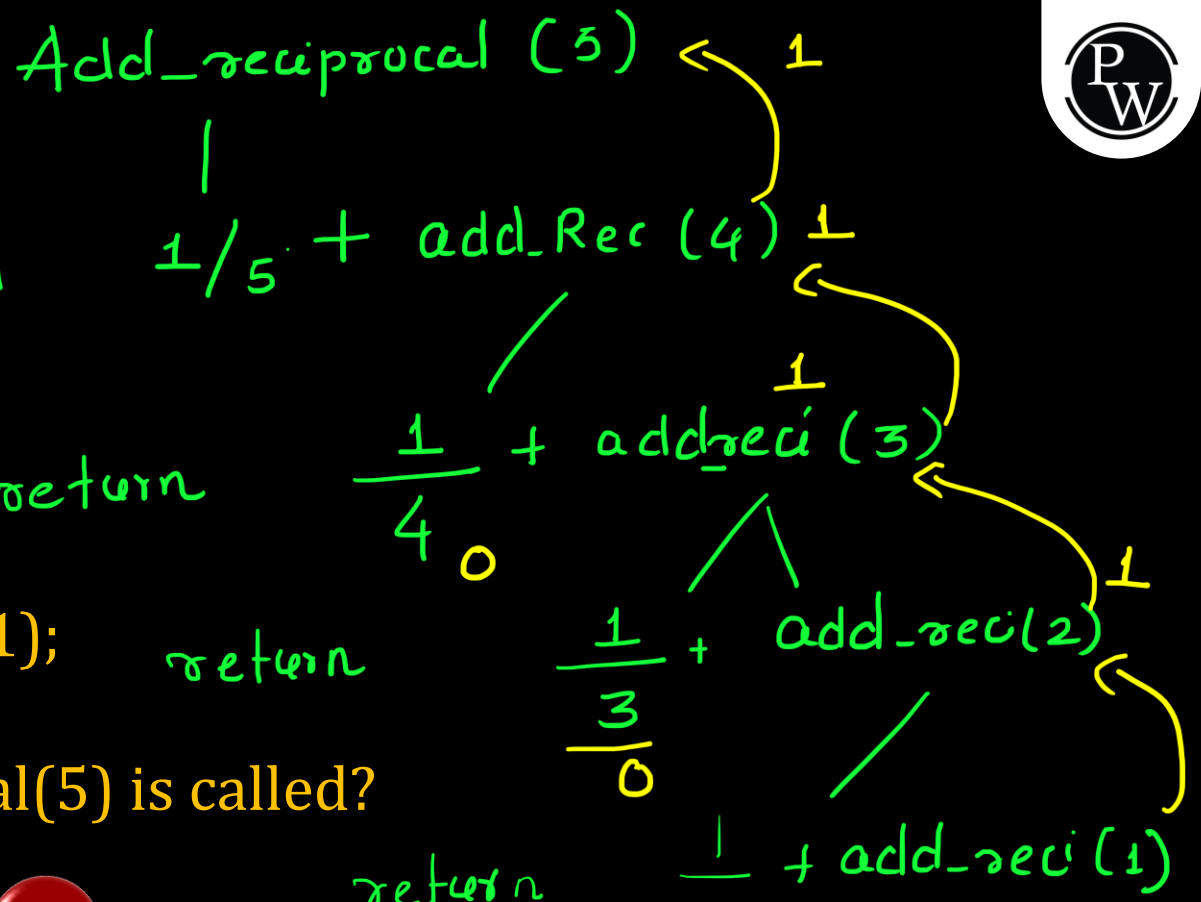
output of the program is 2,3,5,8

Question

#Q. Consider the following function

```
int add_Reciprocals(int n) {
if (n == 1)
        return 1;
else
        return 1 / n + add Reciprocals(n - 1);
}
```

The output the program if add_Reciprocal(5) is called?



(A) 1

(B) 2

(C) 1.617

(D) None of these

[A] Answer

Question

No. of times result called while evaluating result(n)

#Q. Consider the following program

```
int result(int n)
{
if (n= =1)
    return 2;
else
    return 2* result(n-1);
}
```

If n> 0, how many time result will be called to evaluate result(n) (including the initial call)

result(4)
↓
return 2* result(3)
↓
return 2* result(2)
|
return 2* result(1)

2

A 2

B 2^n n:- 4 - 4 times

C n

D 2n result fun called

[c]

Hence Ans [n]

Question

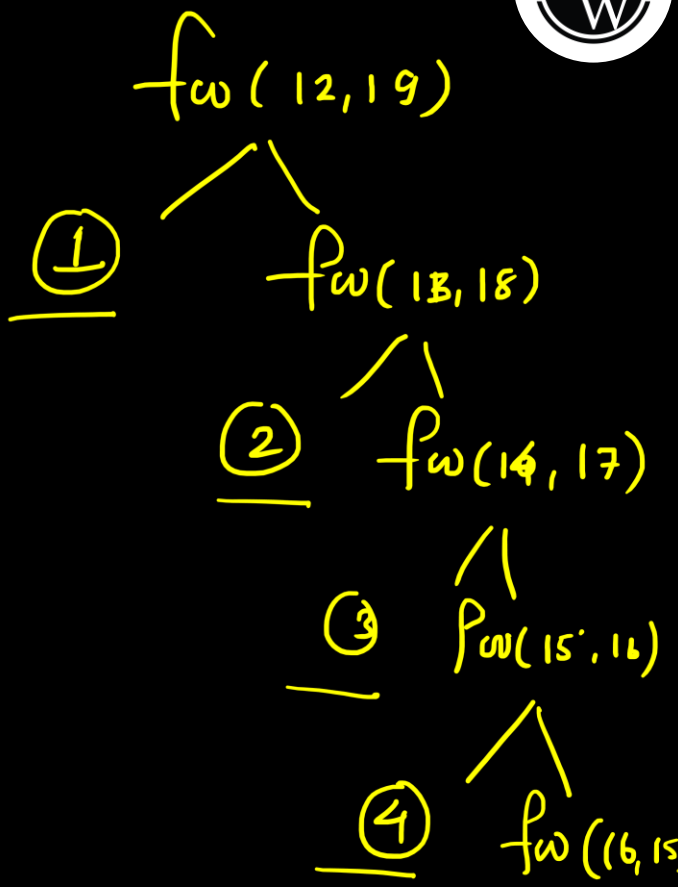
#Q. Consider the following program

```
#include <stdio.h>
void foo(int left, int right) {
    if (left <= right) {
    printf("%d",left);
    foo(left + 1, right-1);
        }
}
int main(){
    foo(12,19) ;
}
```

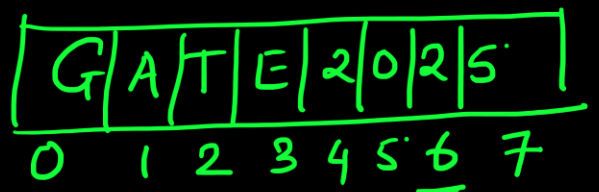
Number of times prinrf statement executed is

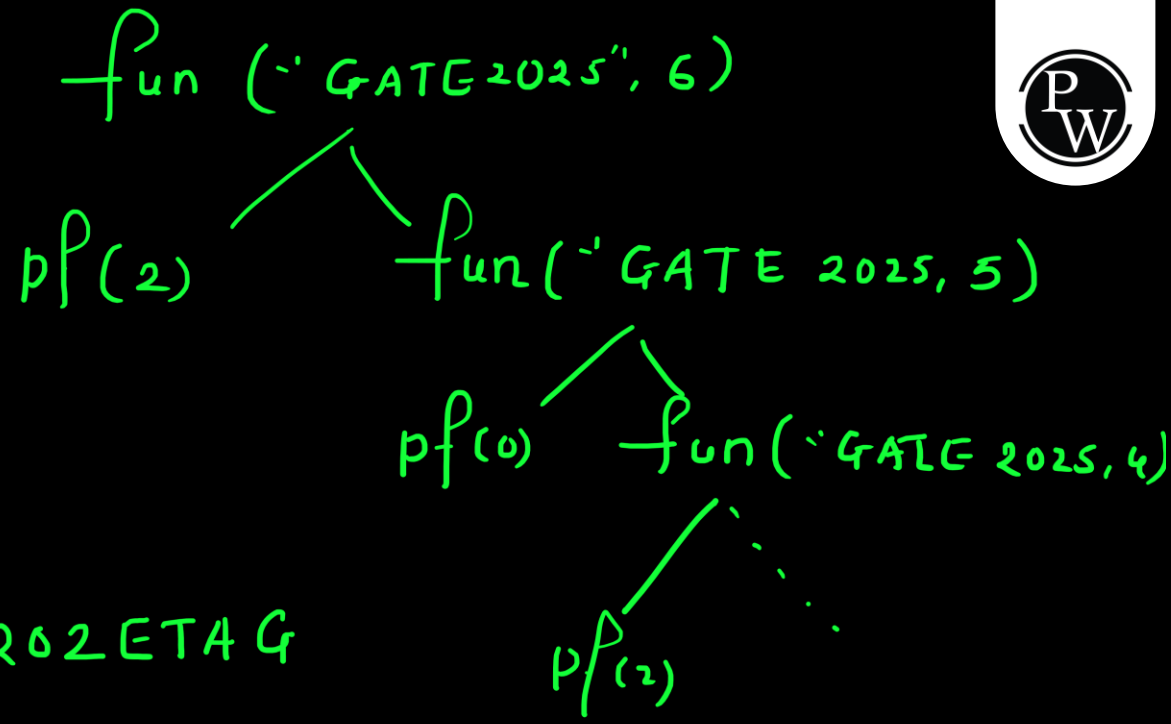
A. 4
B. 5
C. 6
D. 7

[A] - Ans



Question





#Q.

```
#include <stdio.h>
void fun(char * a,int len){
    if(len==0)
        printf("%c",a[len]);
    else {
        printf("%c",a[len]);
        fun(a,len-1);
    }
}
int main(){
    fun("GATE2025", 6) ;
}
```

The output the program us

output : 202ETAG

[A]

A 202ETAG

B ETAG202

C 2025TG

D TG2025

Concept :- partial tree Complete the Recursion tree

Question

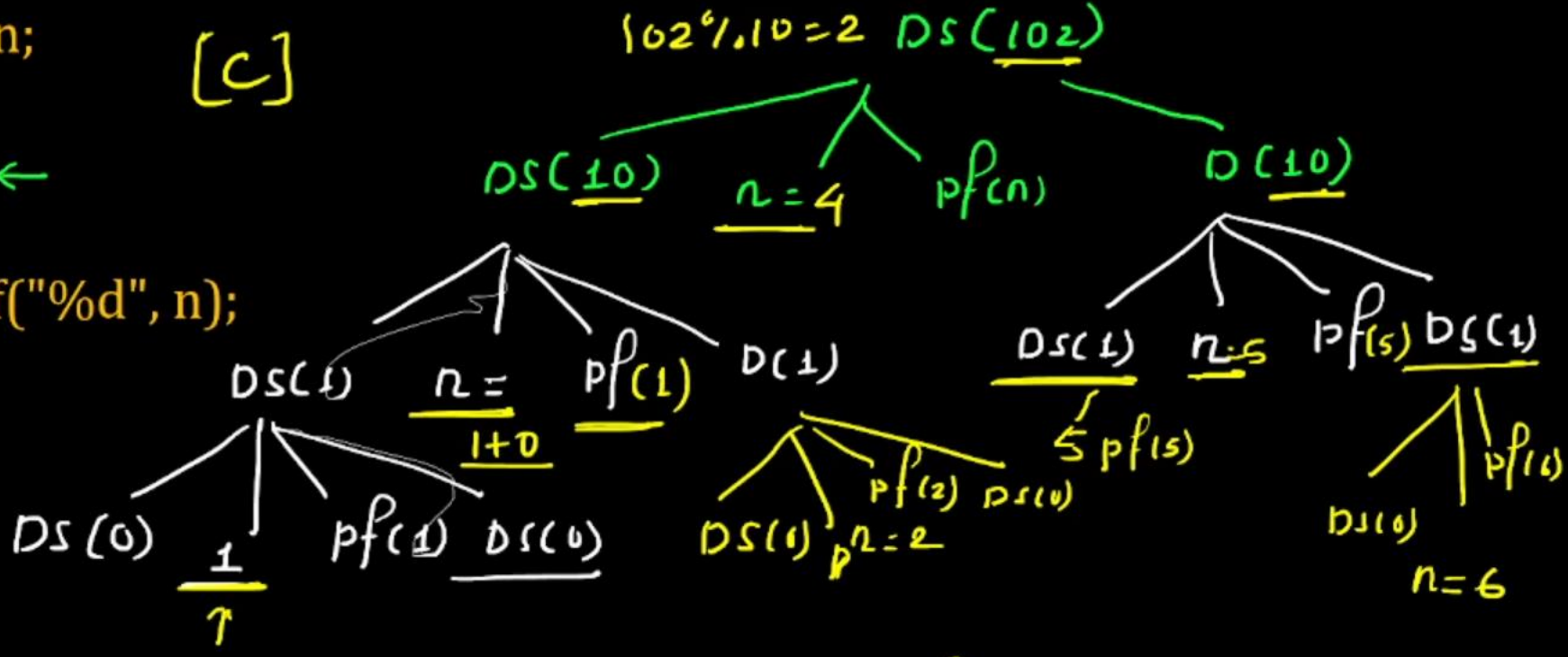
#Q.
```
#include <stdio.h>
void DS( int m){
        static int n;
        if (m==0)
                return;
        DS(m/10);
        n + = m %10;
        if (n!=0)
                printf("%d", n);
        DS(m/10);
        return;
        }
int main() {
DS(102);
}
```

The value printed by the program is _____.

(A) 1124112
(B) 1124211
(C) 1124556
(D) 1124576

static
1/4

[C]



1 1 2 4 5 5 6

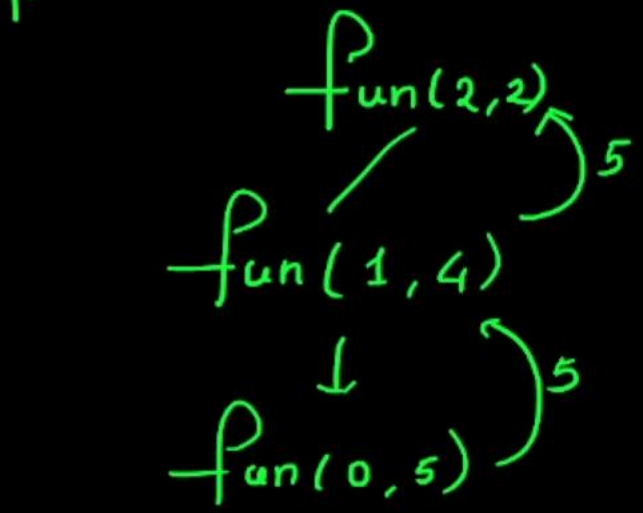
Question

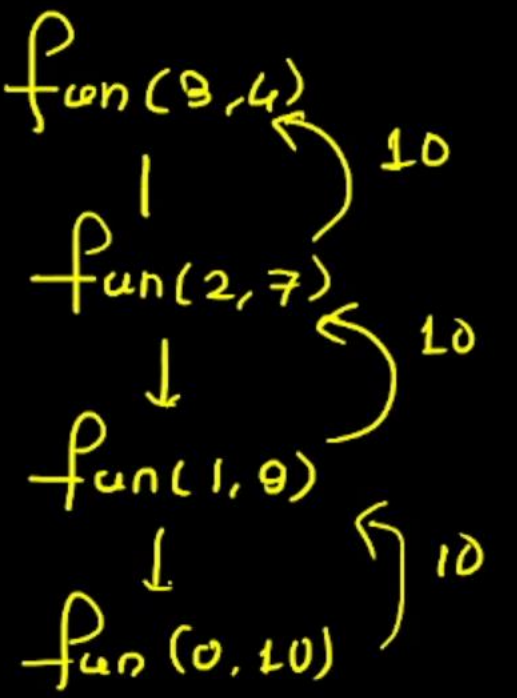
#Q.
```
#include <stdio.h>
int fun(int x, int y) {
    if (x == 0)
        return y;
    else
        return fun(x - 1, x + y);
}
int main() {
    printf("%d", fun(fun(2,2), fun(3,4)));
    return 0;
}
```

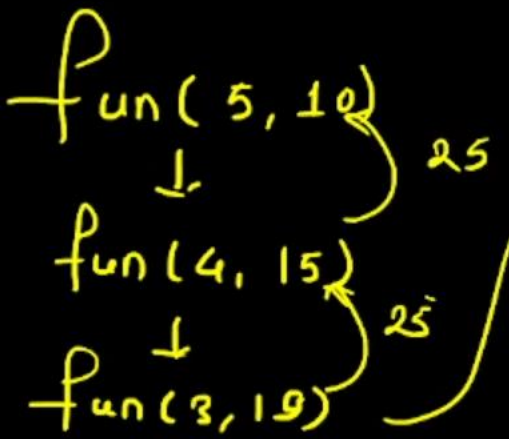
The output of the program is 25

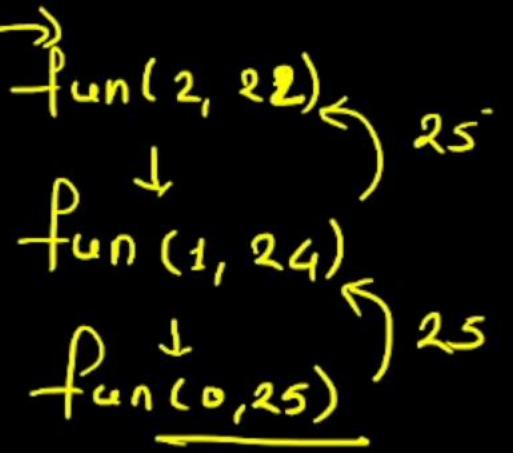
output of program

parameter evaluate









Question

#Q. Consider the following program

```
#include <stdio.h>
int fun1(int n) {
    if (n == 1)
        return 0;
    else
        return 1 + fun1(n / 2);
}
int main(){
    printf("%d", fun1(1024));
    return 0;
}
```

The output of the program is 10 ← 10 is Ang

fun1(1024)
↓ 9
1 + fun1(512) = 10
↓ 8
1 + fun1(256)
↓ 7
1 + fun1(128)
↓ 6
1 + fun1(64)
↓ 5
1 + fun1(32)
↓ 4
1 + fun1(16)
↓ 3
1 + fun1(8)
↓ 2
1 + fun1(4)
↓ 1
1 + fun1(2) ↓ 1 + fun1(1) ↓ 0

= 10

Question

#Q. Consider the following program

```
#include <stdio.h>
int fun1(int n){
   if (n == 1)
      return 0;
   else
      return 1 + fun1(n / 2);
}
int main(){
   printf("%d", fun1(fun1(788)));
   return 0;
}
```

What is the output of the program

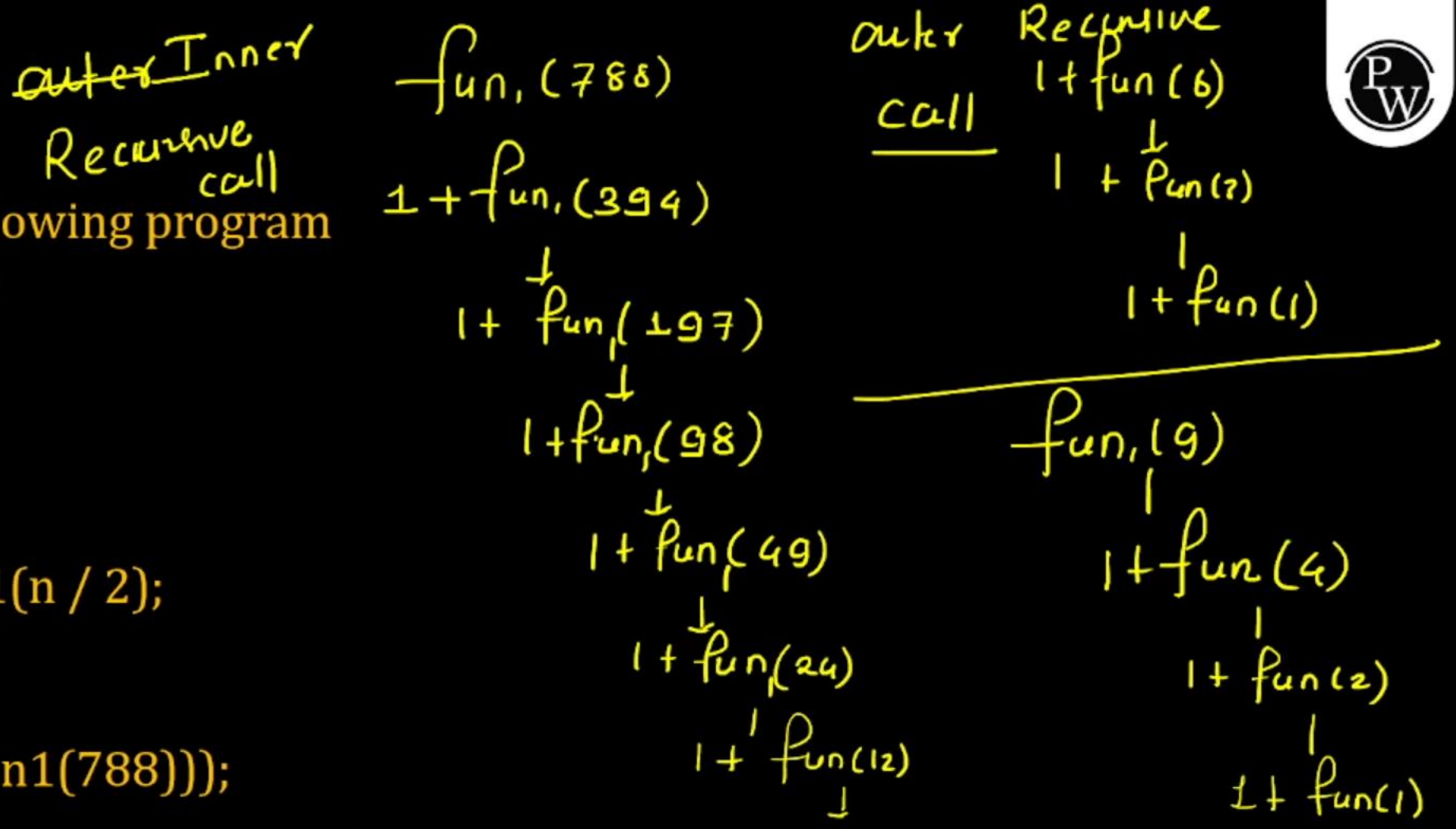
A 5

B 6

C 3

D 4

[C]



THANK - YOU