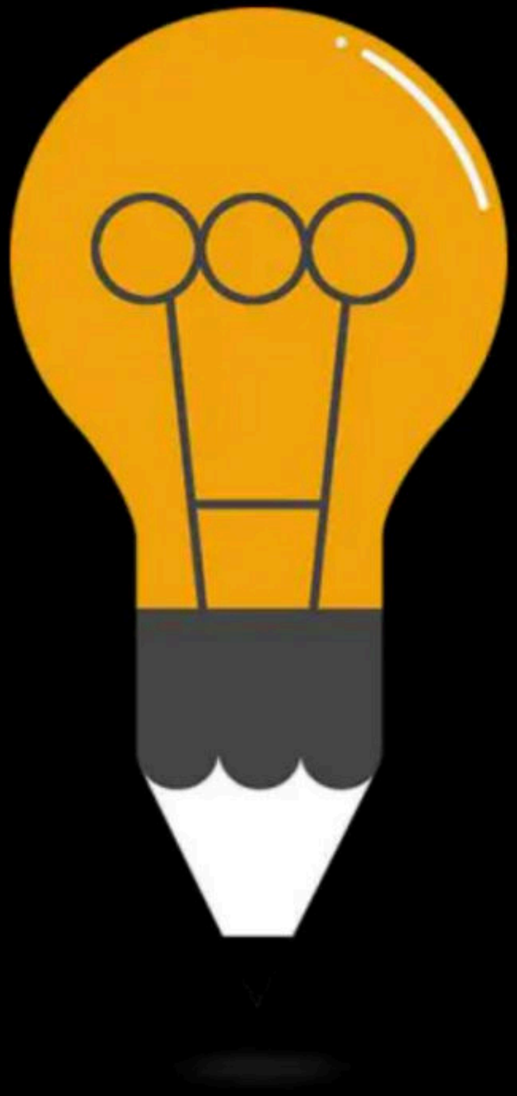
Recursion Questions

Course on C-Programming & Data Structures: GATE - 2024 & 2025

Vishvadeep Gothi • Lesson 14 • Dec 12, 2022



Recursion

By: Vishvadeep Gothi

fun

→ local vs global variables

↳ lifetime ⇒ prog.

↓

lifetime ⇒ function

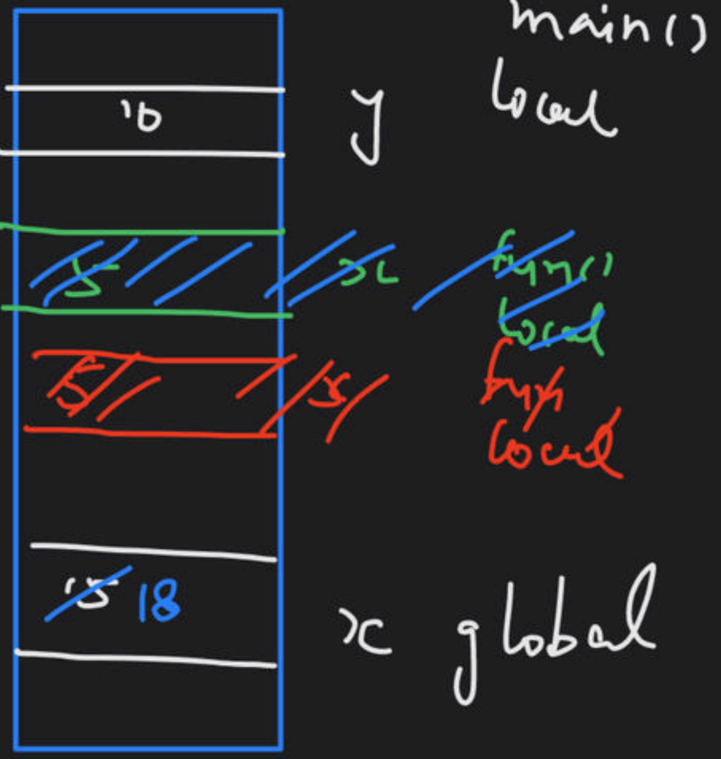
A local variable is created in memory only when the funct^{n} is called and it is deallocated from memory when funct^{n} call is completed.

```
int x;
void fun()
{
  int x = 5;
  printf("%d", x);
}

main ()
{
  int y;
  y = 10;
  x = 15;
  fun();
  x = 18;
  fun();
  printf("%d", x);
}
```

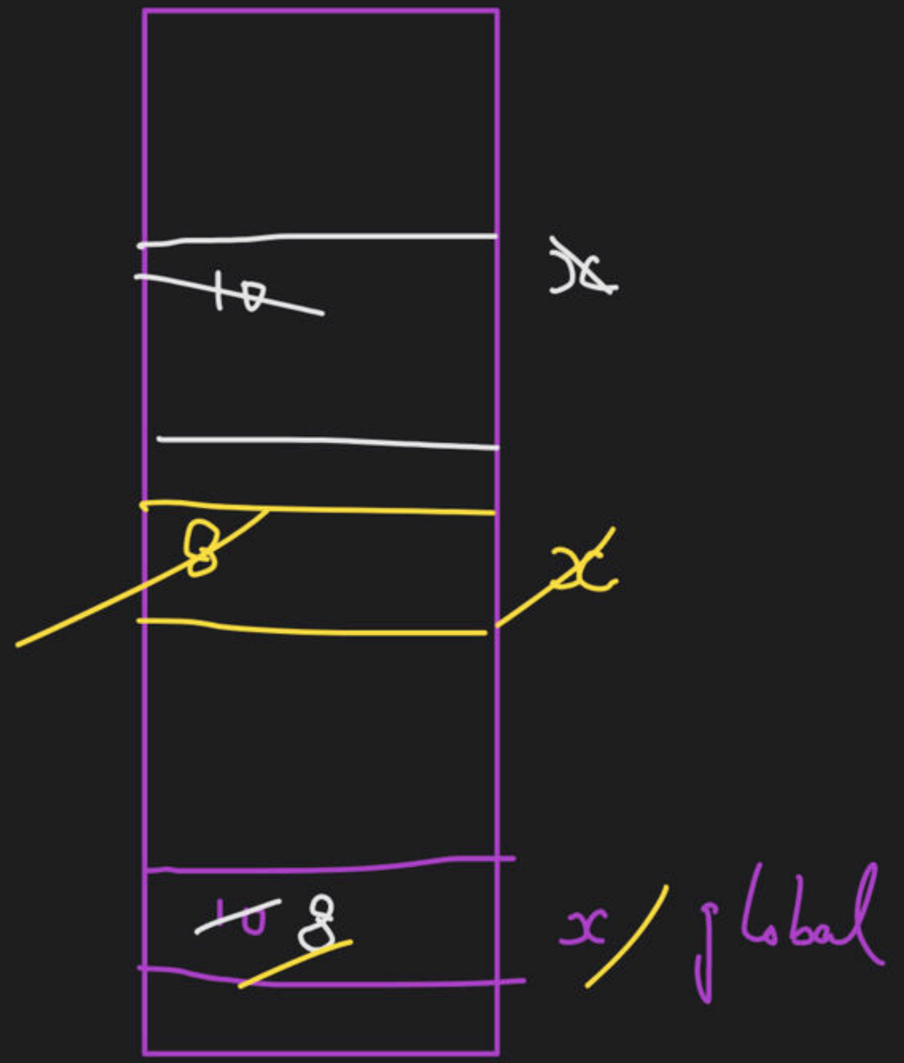
output:-

5
5
18



```
int x;
void fun(int x)
{
 printf("%d\n", x);
}

void main()
 {
    x = 10;
   fun(x);
   x = x - 2;
   fun(x);
}
```



output :-

10

8

Recursion

Funct^{n} calling itself

ex:- function body

```
fun ()
{
  ==
  fun ()
  =
}
```

⇒ A good recursive function has a base condit^{n}, for which funct^{n} does not call itself.

Base / terminat^{n} / exit condit^{n}

⇒ Everytime recursive funct^{n} is called the base condit^{n} should be closer.

⇒ How to solve recursion Questⁿ

⇓

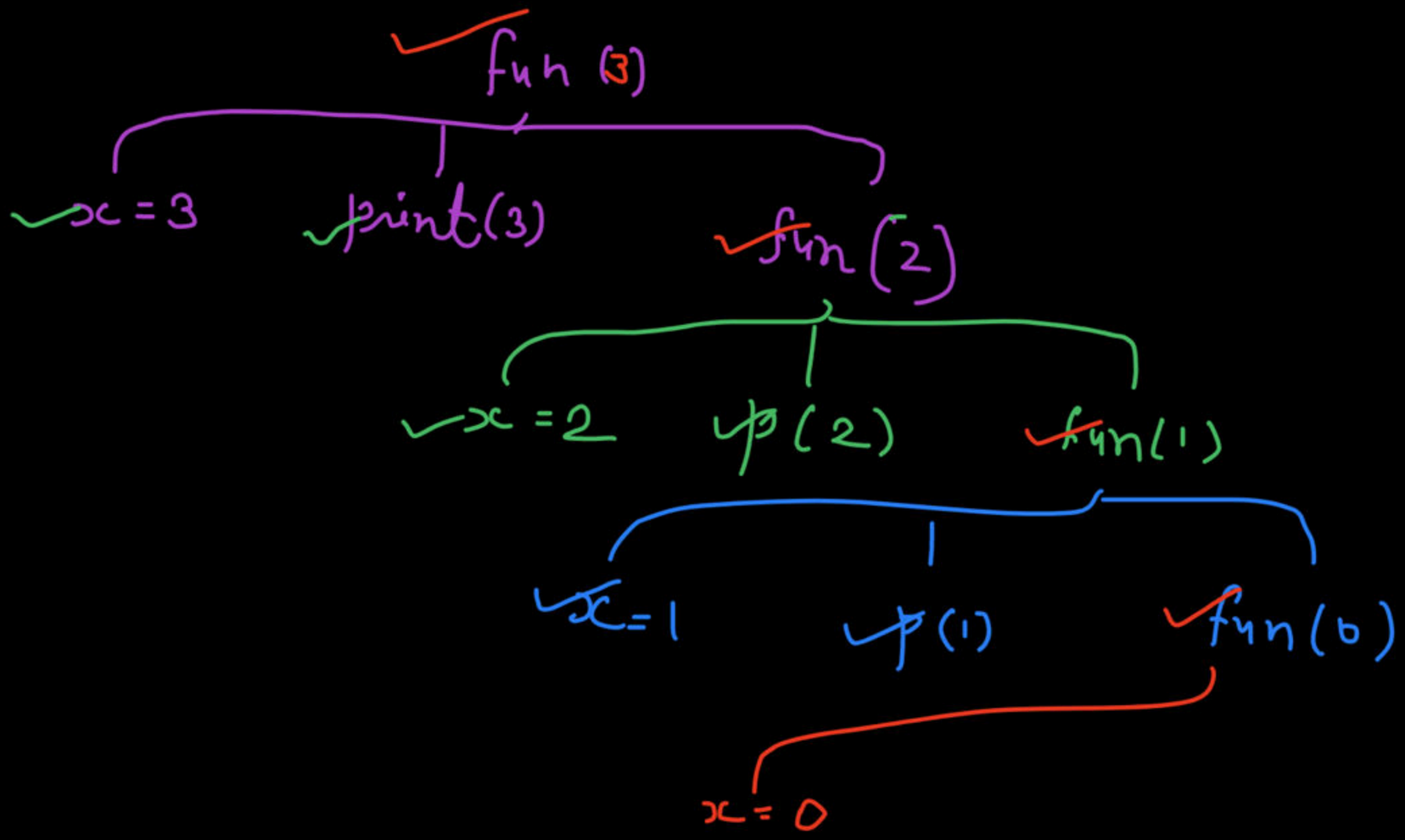
Draw recursion call tree

Question

```
void fun(int x){
  if(x>0)
  {
     printf("%d",x);
     fun(x-1);
  }}
void main() {
  fun(3);
}
```

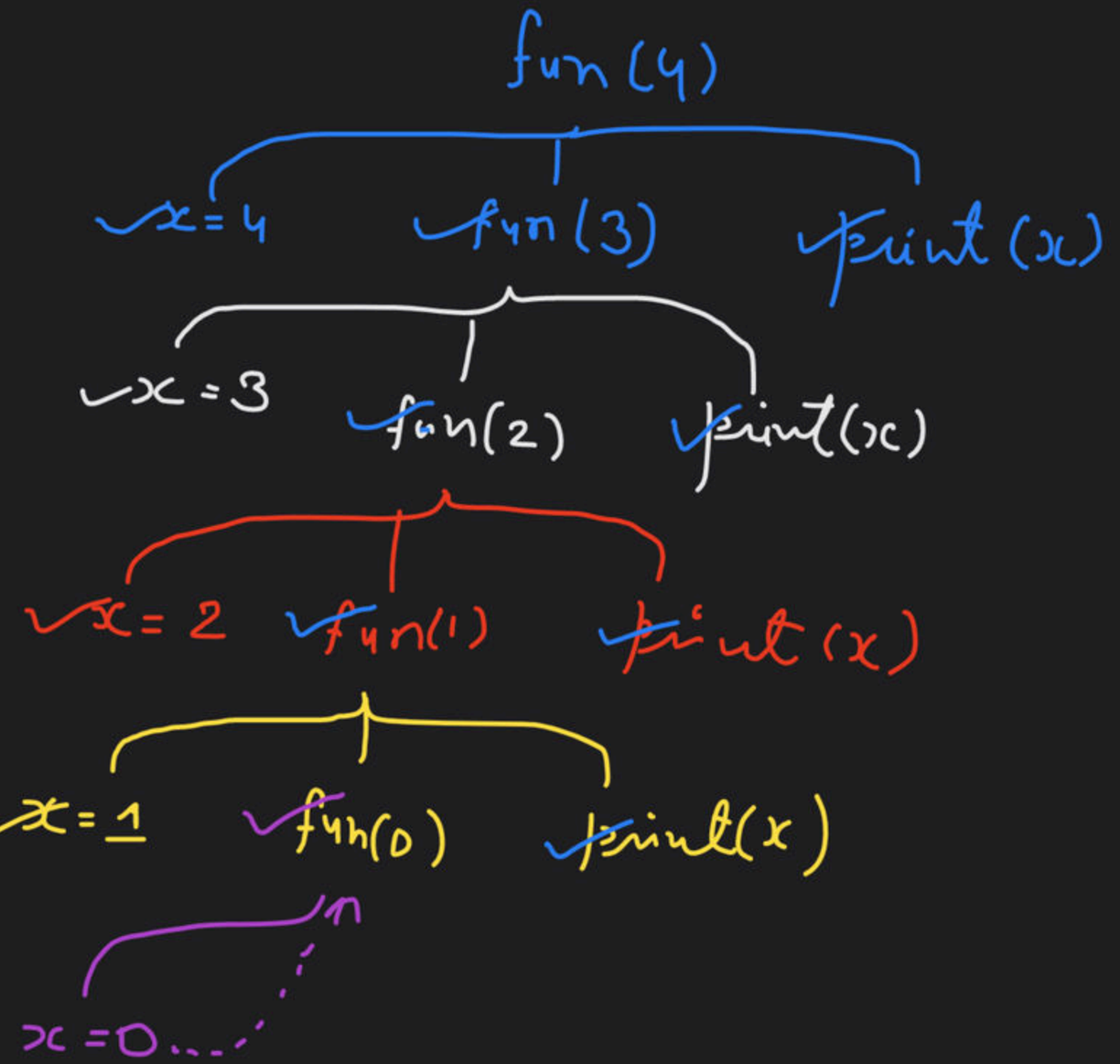
output:-

321



```
void fun(int x)
{
  if (x == 0)
    return;
  fun(x-1);
  printf("%d", x);
}

void main()
{
  fun(4);
}
```

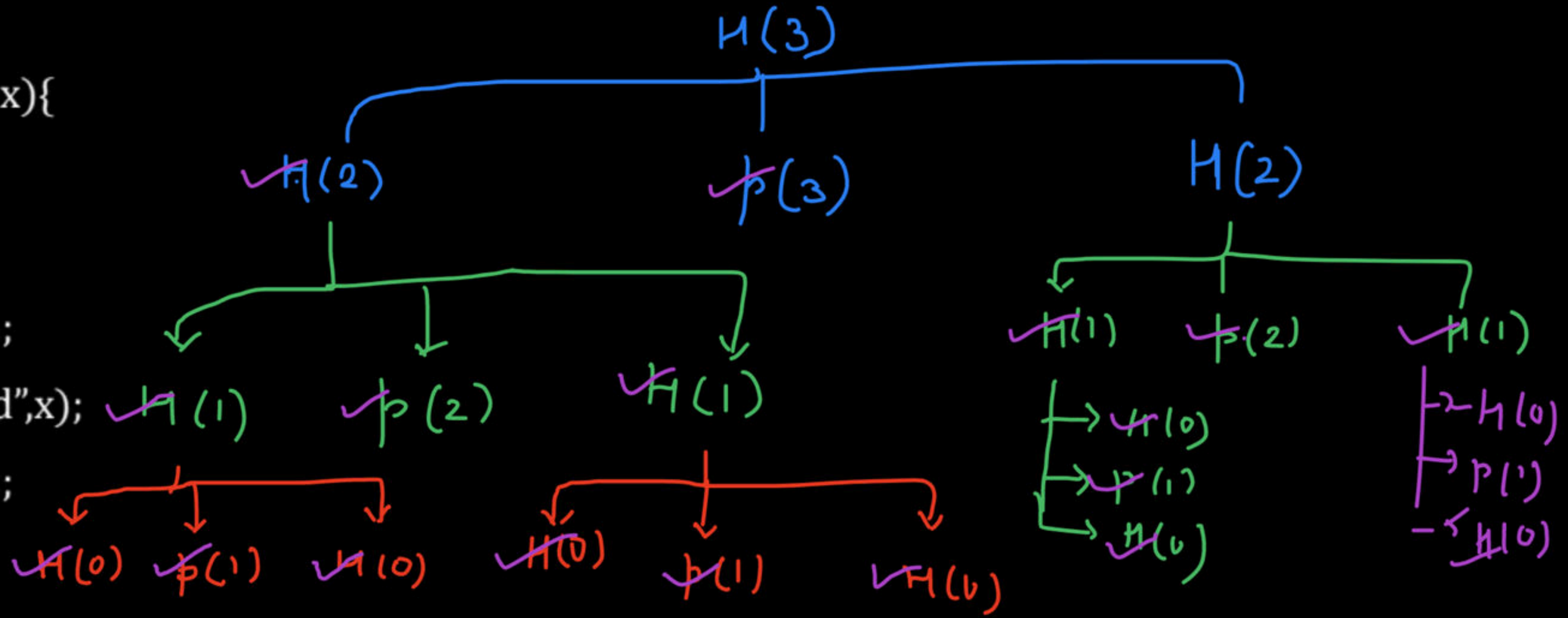


output:- 1234

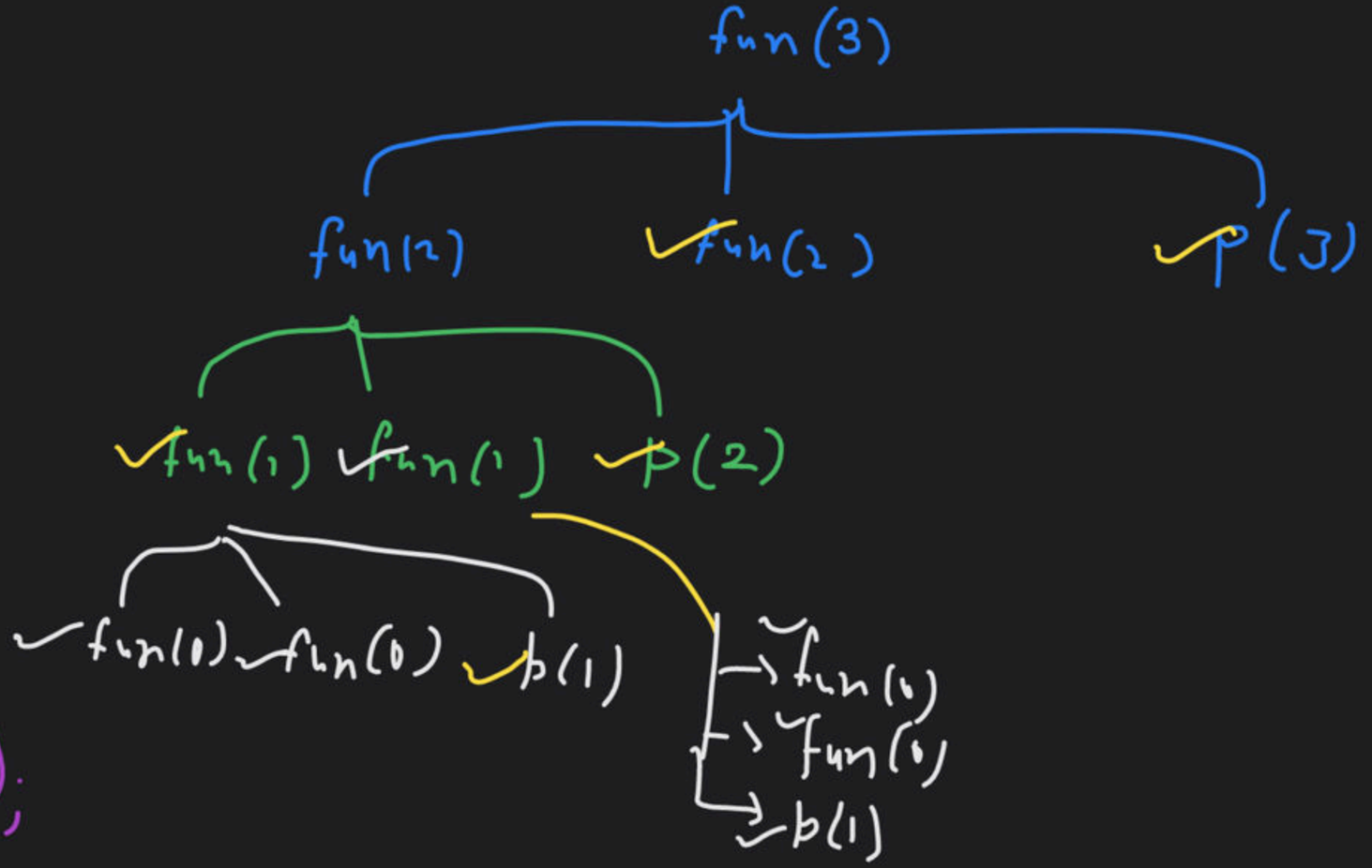
output = 1213121

Question

```
void Head(int x){
  if(x>0)
  {
     Head(x-1);
     printf("%d",x);
     Head(x-1);
  }}
void main() {
  Head(3);
}
```



```
void fun(int n)
{
  if (n>0)
  {
   fun(n-1);
   fun(n-1);
   printf("%d", n);
  }
}
void main()
{
 fun(3);
}
```

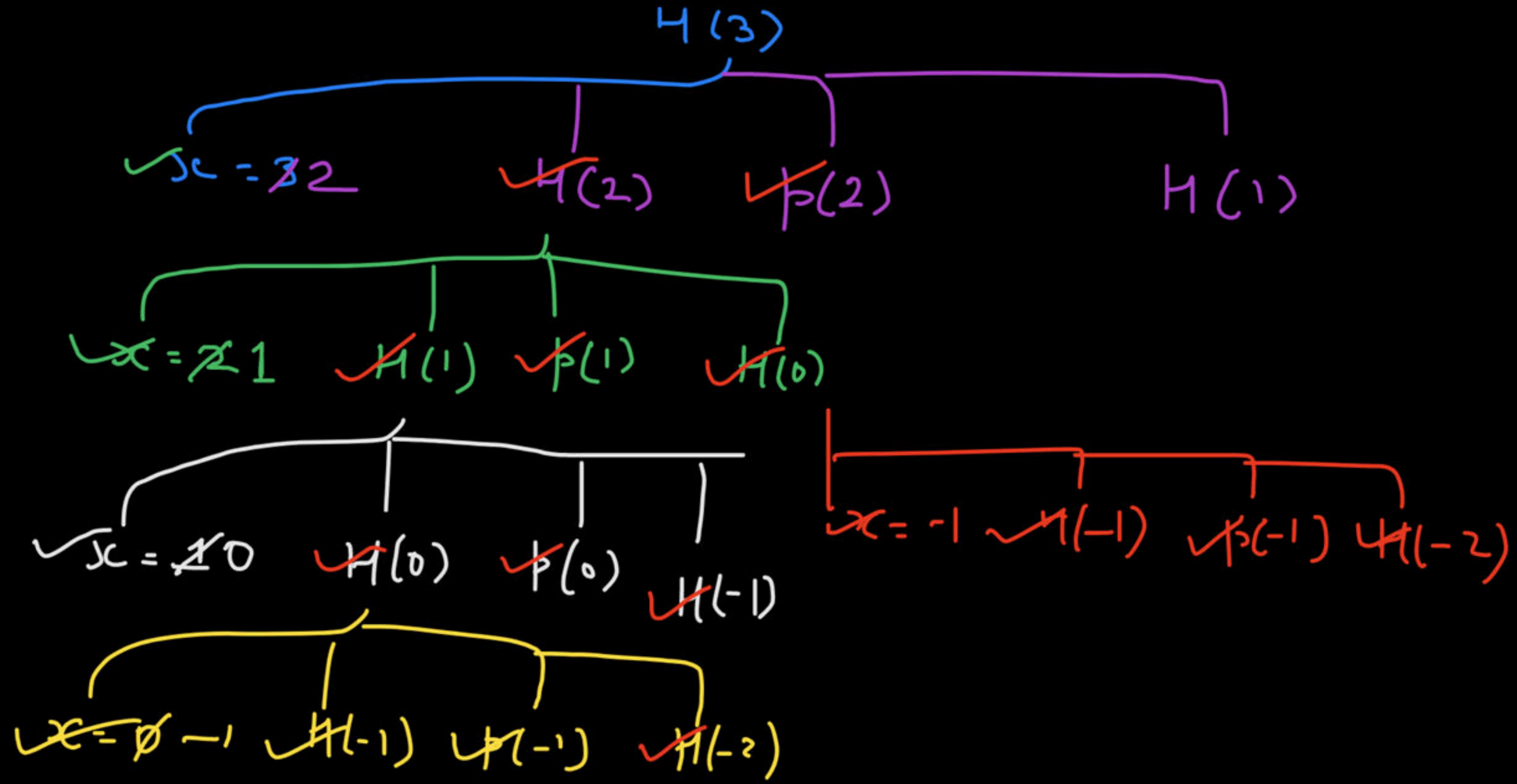


output :- 1 1 2 1 1 2 3

output => 3 2 1 0 -1 -2 -1 0

Question

```
void Head(int x){
  if(x>=0)
  {
    Head(--x);
    printf("%d",x);
    Head(x-1);
  }}
void main() {
  Head(3);
}
```



Question

```
void sample(char *s) {
    if(*s!=NULL)
    {
        sample(s+1);
        sample(s+1);
        printf("%c",*s);
    }}
void main() {
    sample("abc");
}
```

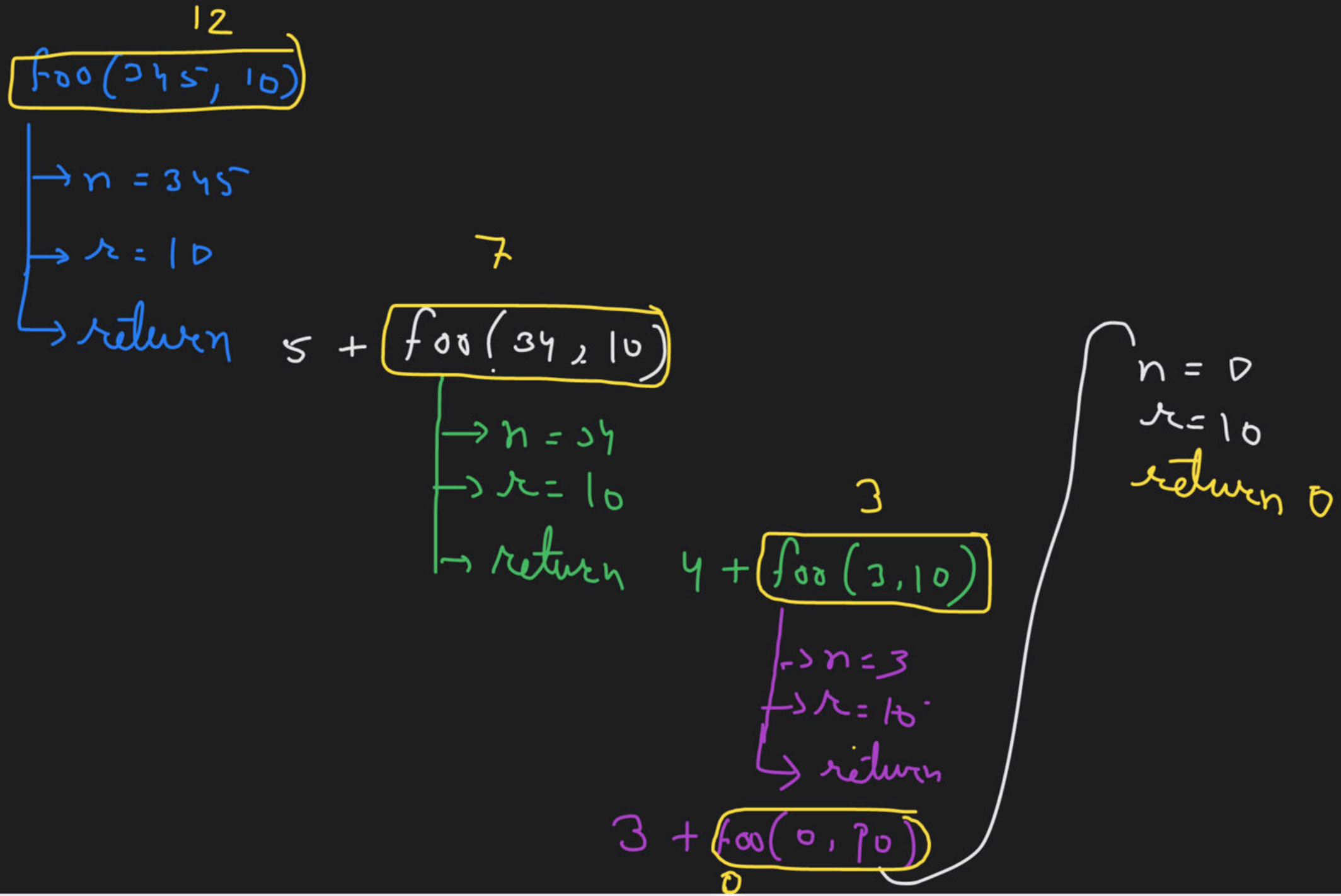
Question

Ans = 12

Consider the following recursive C function that takes two argument.

```
unsigned int foo (unsigned int n, unsigned int r) {
        if (n>0)return ((n%r)+foo (n/r, r));
        else return 0;
}
```

What is the return value of the function foo when it is called as foo (345, 10)?


12
foo(345, 10)
→ n = 345
→ r = 10
→ return 5 +
7
foo(34, 10)
→ n = 34
→ r = 10
→ return 4 +
3
foo(3, 10)
→ n = 3
→ r = 10
→ return
3 +
foo(0, 10)
0
n = 0
r = 10
return 0

Question

Consider the following recursive C function that takes two argument.

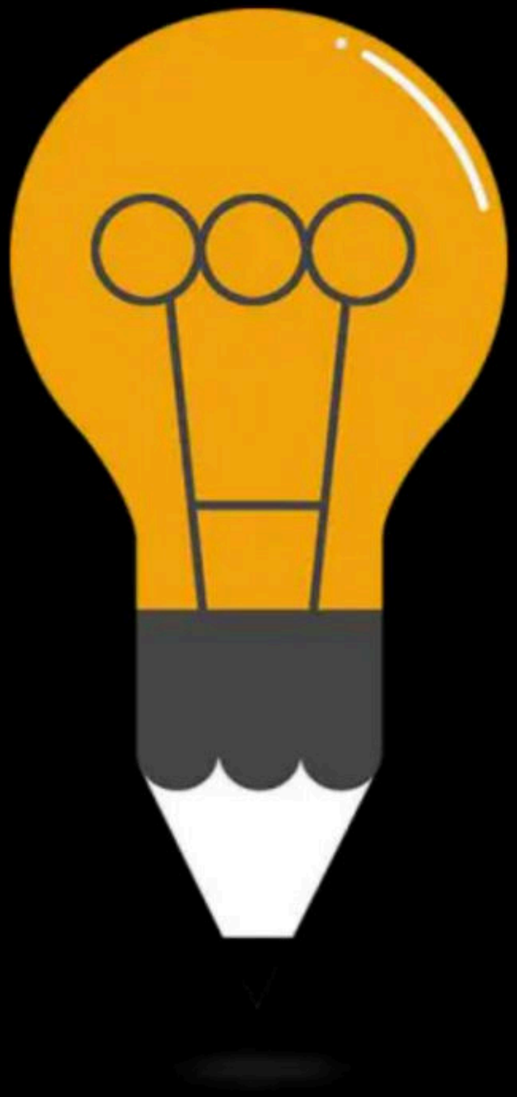
```
unsigned int foo (unsigned int n, unsigned int r) {
        if (n>0)return ((n%r)+foo (n/r, r));
        else return 0;
}
```

What is the return value of the function foo when it is called as foo(513,2) ?

Question

Consider the following C function:

```
int f(int n){
        static int r=0;
        if(n<=0) return 1;
        if(n>3){
      r = n;
      return f(n−2)+2;
        }
        return f(n−1)+r;
}
```



DPP 6

By: Vishvadeep Gothi

Question

Consider the following C function.

```
void convert (int n ) {
        if (n<0)
            printf{“%d”, n);
        else {
            convert(n/2);
            printf(“%d”, n%2);
        }
}
```

Which one of the following will happen when the function convert is called with any positive integer n as argument?

a) It will print the binary representation of n and terminate
b) It will print the binary representation of n in the reverse order and terminate
c) It will print the binary representation of n but will not terminate
d) It will not print anything and will not terminate

Question

Consider the following program written in pseudo-code. Assume that x and y are integers.

```
Count (x, y) {
    if (y !=1 ) {
        if (x !=1) {
            print("*");
            Count (x/2, y);
        }
        else {
            y=y-1;
            Count (1024, y);
        }
    }
}
```

The number of times that the print statement is executed by the call Count(1024, 1024) is?

Question

Consider the following C program:

```
#include <stdio.h>
int counter = 0;
int calc ( int   a,  int  b)  {
int  c;
counter ++;
if (b == 3)   return   (a *  a * a);
else   {
            c = calc (a ,   b/3 );
            return  (c * c* c );
}}

int  main () {
            calc (4,  81)
            printf ( "%d",  counter ); }
```

The output of this program is _____ ?

Question

Consider the following C program:

```
#include <stdio.h>
int counter = 0;
int calc ( int   a,  int  b)  {
int  c;
counter ++;
if (b == 3)   return   (a *  a * a);
else   {
            c = calc (a ,   b/3 );
            return  (c * c* c );
}}

int  main () {
            calc (4,  81)
            printf ( "%d",  counter ); }
```

The output of this program is _____ ?

Happy Learning.!