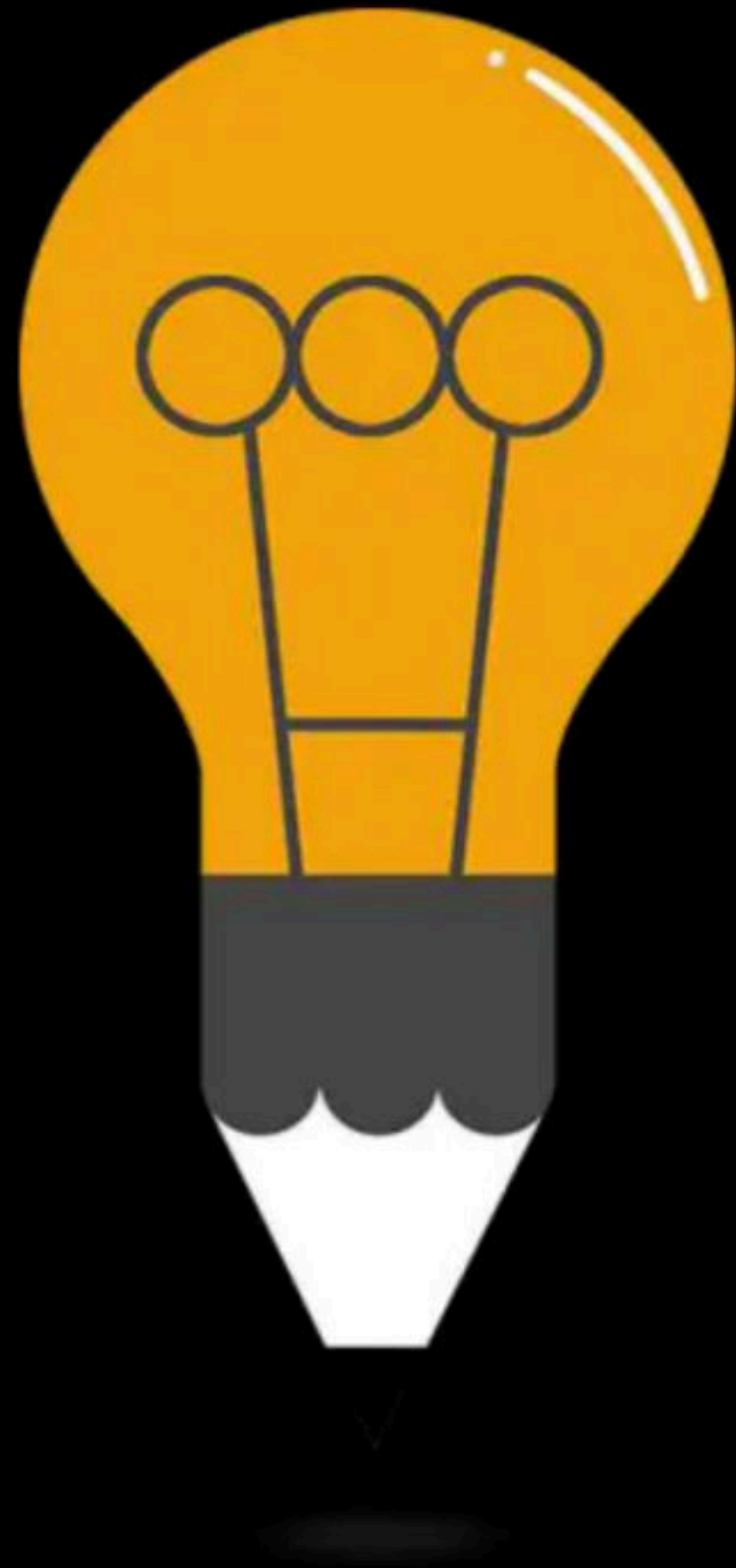




Storage Classes

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



Recursion & Storage Classes

By: Vishvadeep Gothi

Recursion

Question

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

Question

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

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

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)?

Ans = 12

$(345)_{10} \Rightarrow 3+4+5 = 12$

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) ? *Ans = 2*

$$\begin{aligned} (513)_{10} &= (100000001)_2 \\ &\quad \downarrow \\ &1+0+0+0+0+0+0+0+0+1 \\ &= 2 \end{aligned}$$

$\boxed{\text{foo}(513, 2)}^2$

|
return $1 + \boxed{\text{foo}(256, 2)}$

|
return $0 + \text{foo}(128, 2)$

|
 $0 + \text{foo}(64, 2)$

|
 $0 + \text{foo}(32, 2)$

|
 $0 + \text{foo}(2, 2)$

|
 $0 + \text{foo}(1, 2)$

|
 $1 + \boxed{\text{foo}(0, 2)}$

$$2^0 = 1$$

$$2^1 = 2$$

$$4$$

$$8$$

$$16$$

$$32$$

$$64$$

$$128$$

$$256$$

$$512$$

Ans = 91

Question

What value would the following function return for fun(95)?

```
fun(x){  
  if(x>100) return x-10;  
  else return fun(fun(x+11));  
}
```

fun(95) → fun(fun(106))
96

return fun(fun(107))
97

return fun(fun(108))
98

return fun(fun(109))
99

return fun(fun(110))
100

return fun(fun(111))
101

91


```
int fun(int n)
```

```
{
```

```
    if (n < 3)
```

```
        return 1;
```

```
    else
```

```
        return fun(n-1) + fun(n-3) + 1
```

```
}
```

7 fun(5)

return ⁵fun(4) + ¹fun(2) + 1

return ³fun(3) + ¹fun(1) + 1

return ¹fun(2) + ¹fun(0) + 1

returned value for fun(5) = ?

Ans = 7

in this function $\text{fun}()$ \Rightarrow returned value
of $\text{fun}(n)$ = no. of functⁿ
calls for
execution of $\text{fun}(n)$
($n \geq 3$)

Ques]

no. of times $\text{fun}()$ is called for calculating $\text{fun}(\text{fun}(5))$?

Ans = 24

Solⁿ

$$\text{fun}(\text{fun}(5)) \quad 17$$

7

$$\text{return } \text{fun}(6) + \text{fun}(4) + 1$$

" 5

$$\text{return } \text{fun}(5) + \text{fun}(3) + 1$$

7 3

first call $\text{fun}(5) \Rightarrow 7$ fun calls

call $\text{fun}(7) \Rightarrow 17$ fun calls

24 fun calls

static variable

In recursion, it is created for first call and used by all recursion calls as a single copy.

By default it is initialized by zero.

Ans = 18

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;  
}
```

```
void main()  
{
```

```
    printf("%d", f(5));  
}
```

Diagram illustrating the recursive calls for $f(5)$:

- $f(5)$ (boxed) returns 18. It is calculated as $f(3) + 2$.
- $f(3)$ returns 11. It is calculated as $f(2) + r$.
- $f(2)$ returns 6. It is calculated as $f(1) + r$.
- $f(1)$ returns 1. It is calculated as $f(0) + r$.
- $f(0)$ (boxed) returns 1.

$r = 5$

Question

Consider the following C function:

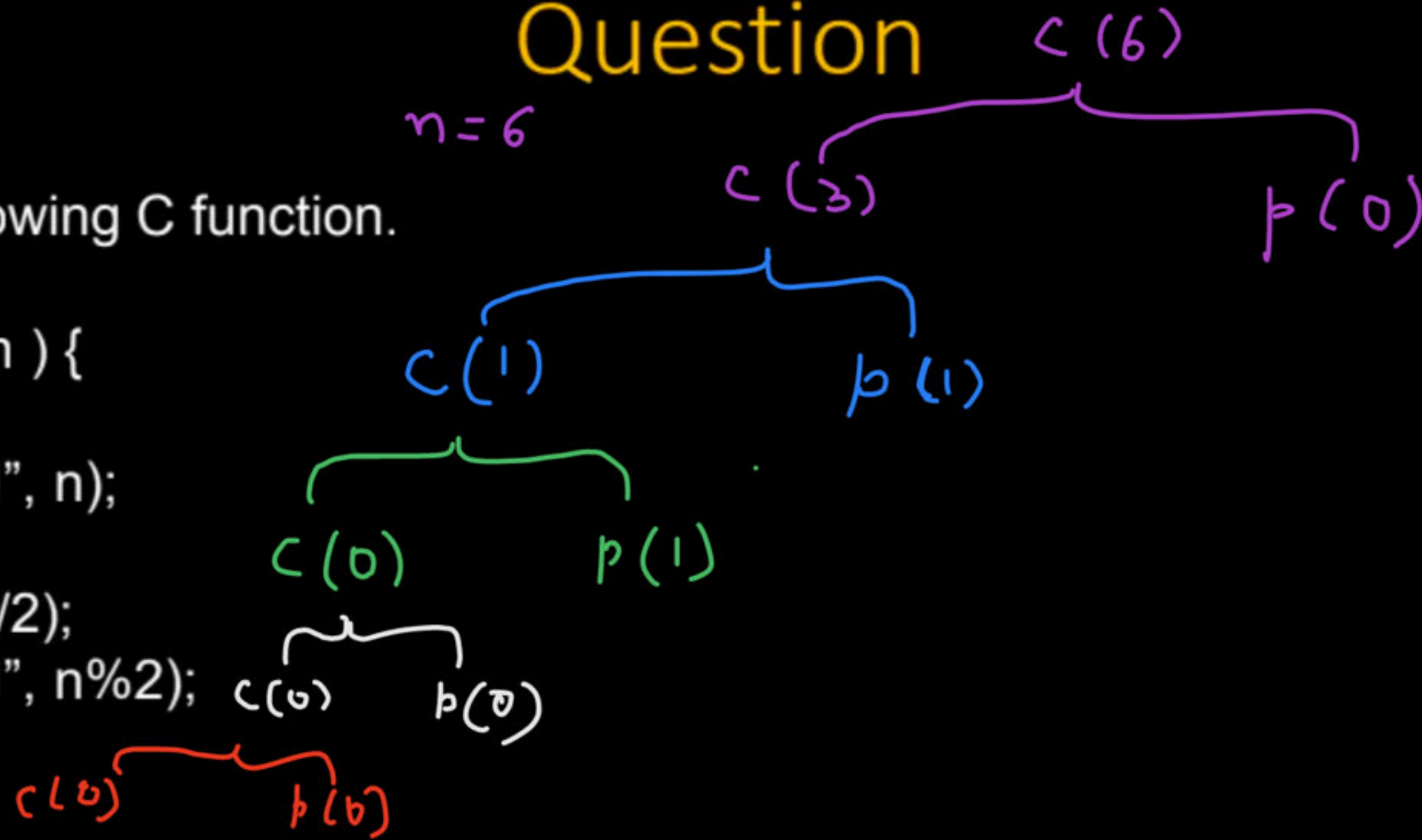
```
int funf(int n){  
    int x=1, k;  
    if(n==1) return x;  
    for(k=1; k<n; k++)  
        x=x+fun(k)*fun(n-k);  
    return x;  
}
```

The return value of fun(5) is _____?

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

Ans = 10230

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);  
    }  
  }  
}
```

count (1024, 1024)

count (512, 1024)

256, 1024

128, 1024

64

32

16

8

4 2 count (1, 1024)

x = 1024

y = 1024

*

*

*

*

*

*

10 times

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

count (1024, 1024)

|

print * times

count (1024, 1023)

|

print * times

count (1024, 1022)

⋮

count (1024, 1)

for $y = 1024$ 10 times

1023

10 times

⋮

2

10 times

1023 times

10 times

Ans = 1023 * 10

= 10230

Question

counter = 0 1 2 3 4

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 4 ?

calc (4, 81)
/

c = calc (4, 27) — c = calc (4, 9);
return (c * c * c) return (c * c * c);

c = calc (4, 3) — 4 * 4 * 4
return (c * c * c)

$((4^3)^3)^3$

Happy Learning.!



storage class
macro
constant
string
pointers