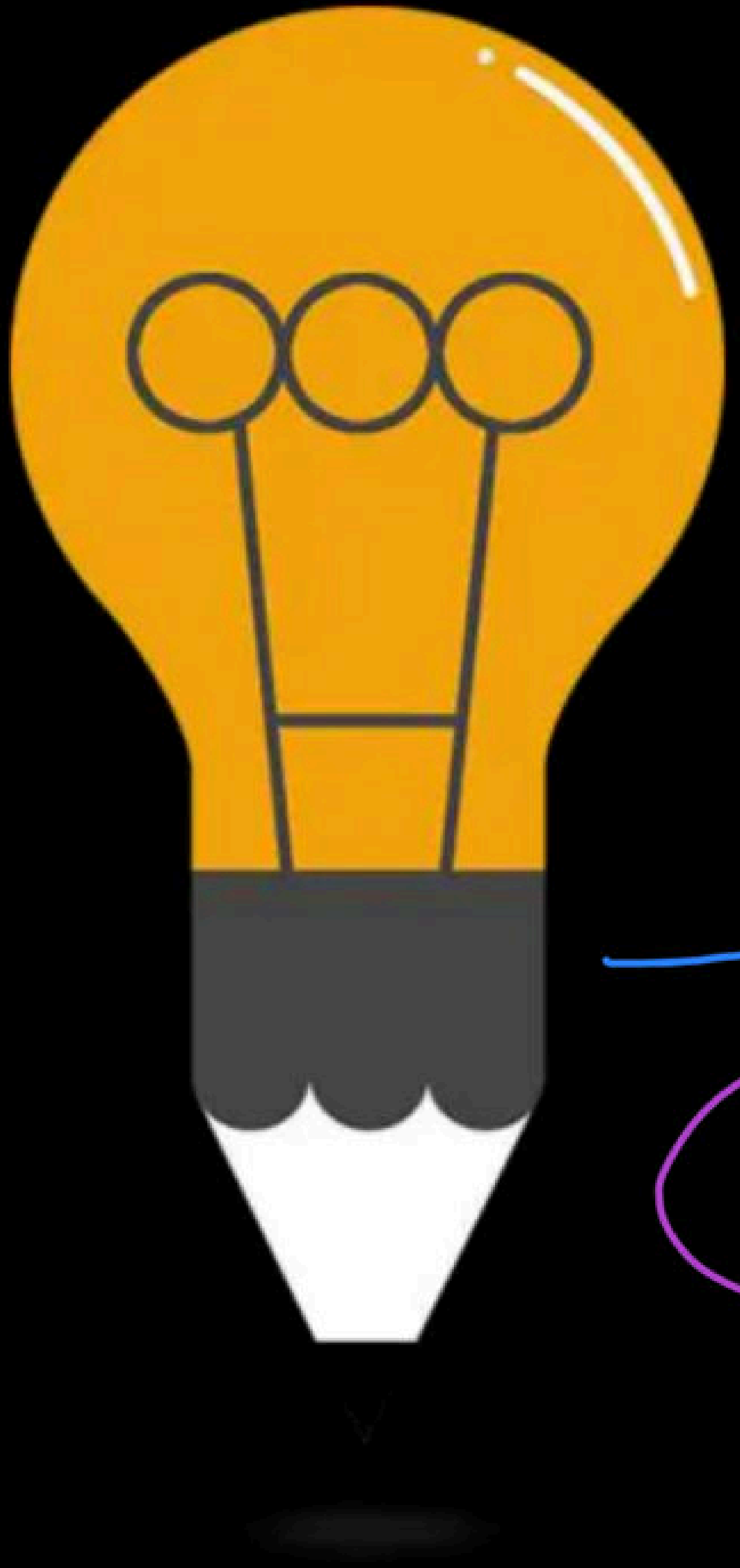
Recursion

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

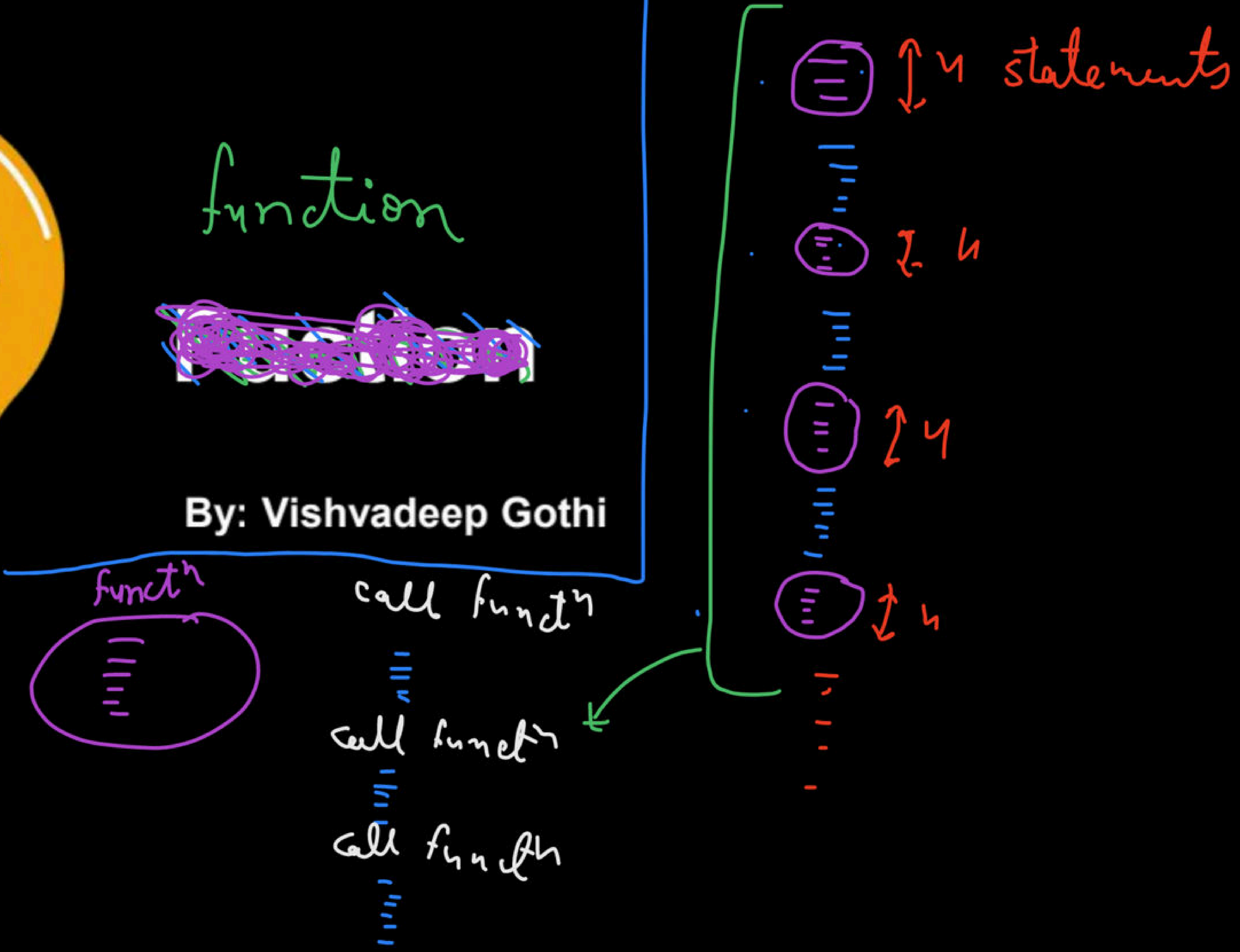
Vishvadeep Gothi • Lesson 12 • Dec 8, 2022



function

~~Function~~

By: Vishvadeep Gothi



Function

- → funct^n declarat^n
- → funct^n definit^n (body)
- → funct^n call

funct^n declarat^n
- → ① Name
- → ② set of inputs ⇒ arguments or paramete
- → ③ output ⇒ Return value

ex:- funct^n declarat^n

```
return_type name (inputtype1, inputtype2, --);
float fun(int, int, float);
```

functⁿ body :-

ex :-

```
float fun (int x, int y, float z)
{                      int x = 15, int y = 10, float z = 3.6
   float abc;
                                   abc =
   abc = x * y * z;                 540
   return abc;
}
```

functⁿ call :-

```
         540
f = fun (15, 10, 3.6)
```

$$f = \boxed{fun(15, 10, 3.6)}$$

If no any input in functⁿ

declaratⁿ :- returntype name()

or

body

or

returntype name (void)

If no output or return value

void name ()

```
#include <stdio.h>
float area (float);

float area ( float r)
{
  float a;
  a = 3.14 * r * r;
  return a;
}
```

float r = radius

r = 1.2

4.5216

```
return (3.14 * r * r)
```

radius = 1.2

```
void main()
{
 float radius;
 scanf( "%f", &radius);
 printf("Area of circles is = %f", area (radius));
}
```

4.5216

4.521606

Parameter Passing

$$\begin{tikzcd}
\longrightarrow \text{value of variable} & \text{funct}^{n}\text{ call by value} \arrow[l, Leftarrow] \\
\longrightarrow \text{Address of variable} & \text{funct}^{n}\text{ call by address} \arrow[l, Leftarrow]
\end{tikzcd}$$

call by value:-

```
void xyz(int x)
{
  x = 5;
}

void main()
{
  int a = 10;
  printf("%d\n", a);
  xyz(a);
  printf("%d", a);  }
```

int x = a;

x = ~~10~~ 5

a = 10

output:-

10

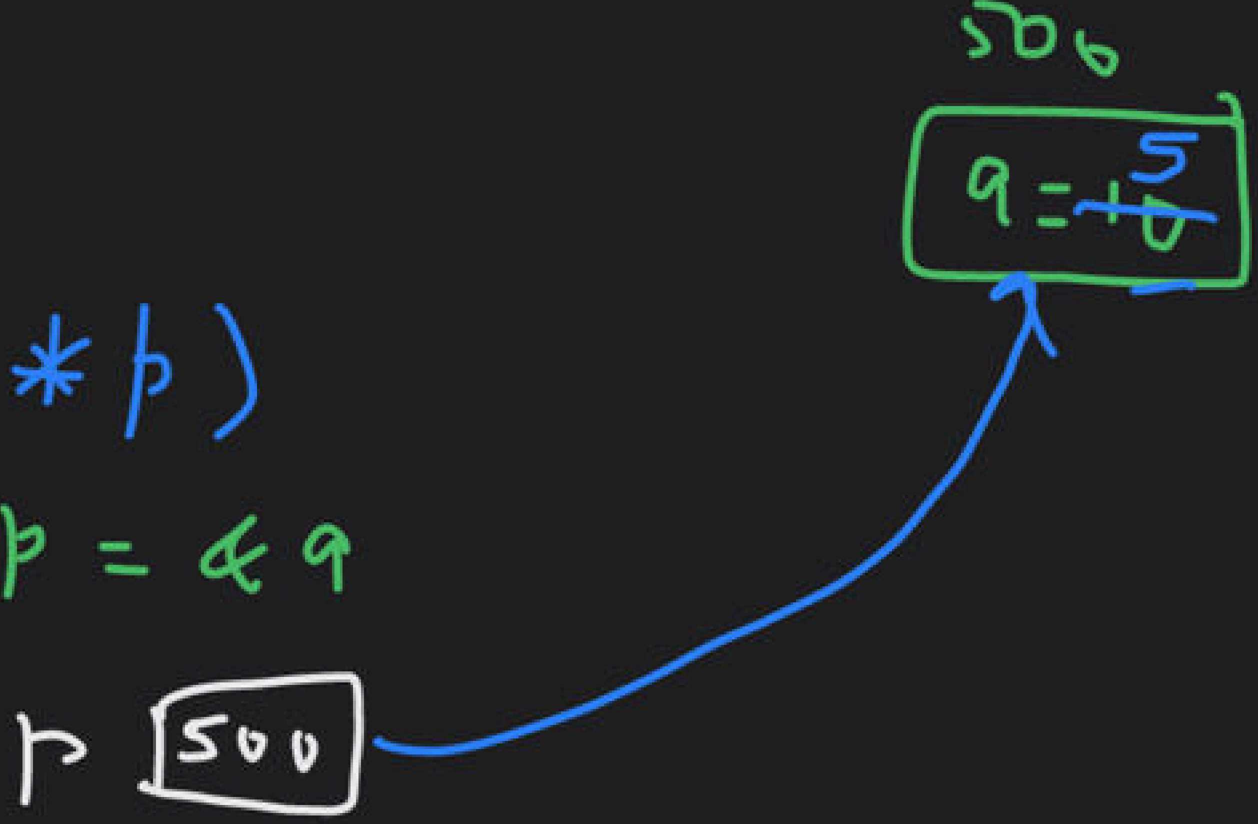
10

call by address:-

```
void fun (int *);
```

```
void fun (int *p)
{
  *p = 5;
}
```

int *p = &a



```
void main()
{
 int a = 10;
 printf("%d\n", a);
 fun(&a);
 printf("%d\n", a);
}
```

output:-
```
10
5
```

Global vs Local Variable

Global:

- → declared outside all funct^ns
- → visible from all funct^ns

Local:

- → declared within a funct^n
- → visible from it's own funct^n

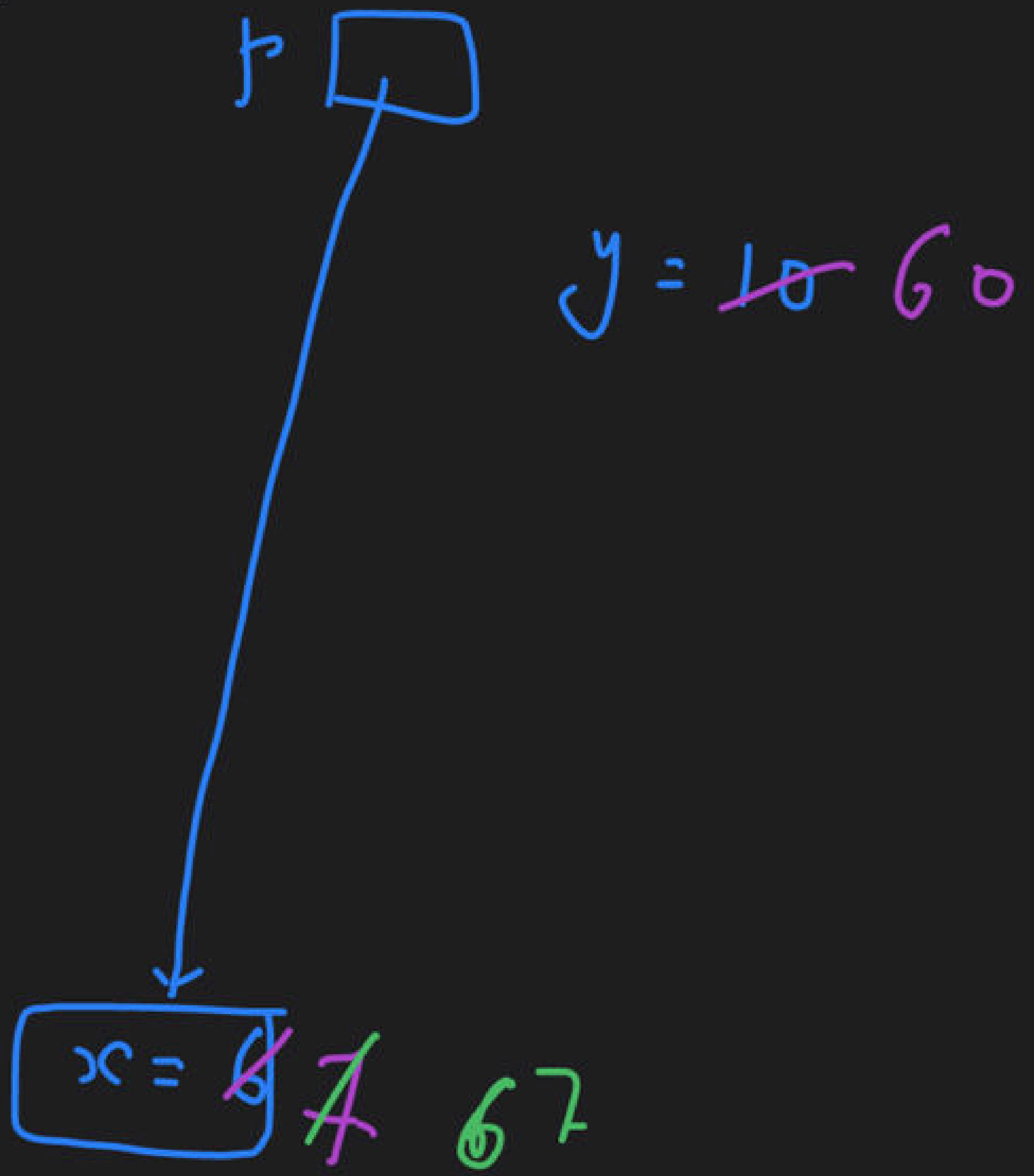
```
#include <stdio.h>

int x = 5;

void fun1()
{
  int x =10;
  printf("%d\n", x);
}

void fun2()
{
  printf("%d\n", x);
}

void main()
{
  printf("%d\n", x);
  fun2();
  fun1();
  printf("%d\n", x);
}
```

output:-

```
5
5
10
5
```

```
void fun()
{
  int x = 10;
  printf("%d", x);
  printf("%d", y);   ← error
}
void main()
{
  int y = 5;
  fun();
  printf("%d", x);   ← error
}
```

```
void fun( int *p)
{
   int y = 10;
   y = y * (*p);
   (*p) ++;
   *p = y + (*p);
}

void main()
{
   int x = 6;
   fun (&x);
   printf("%d", x);
}
```



output => 67

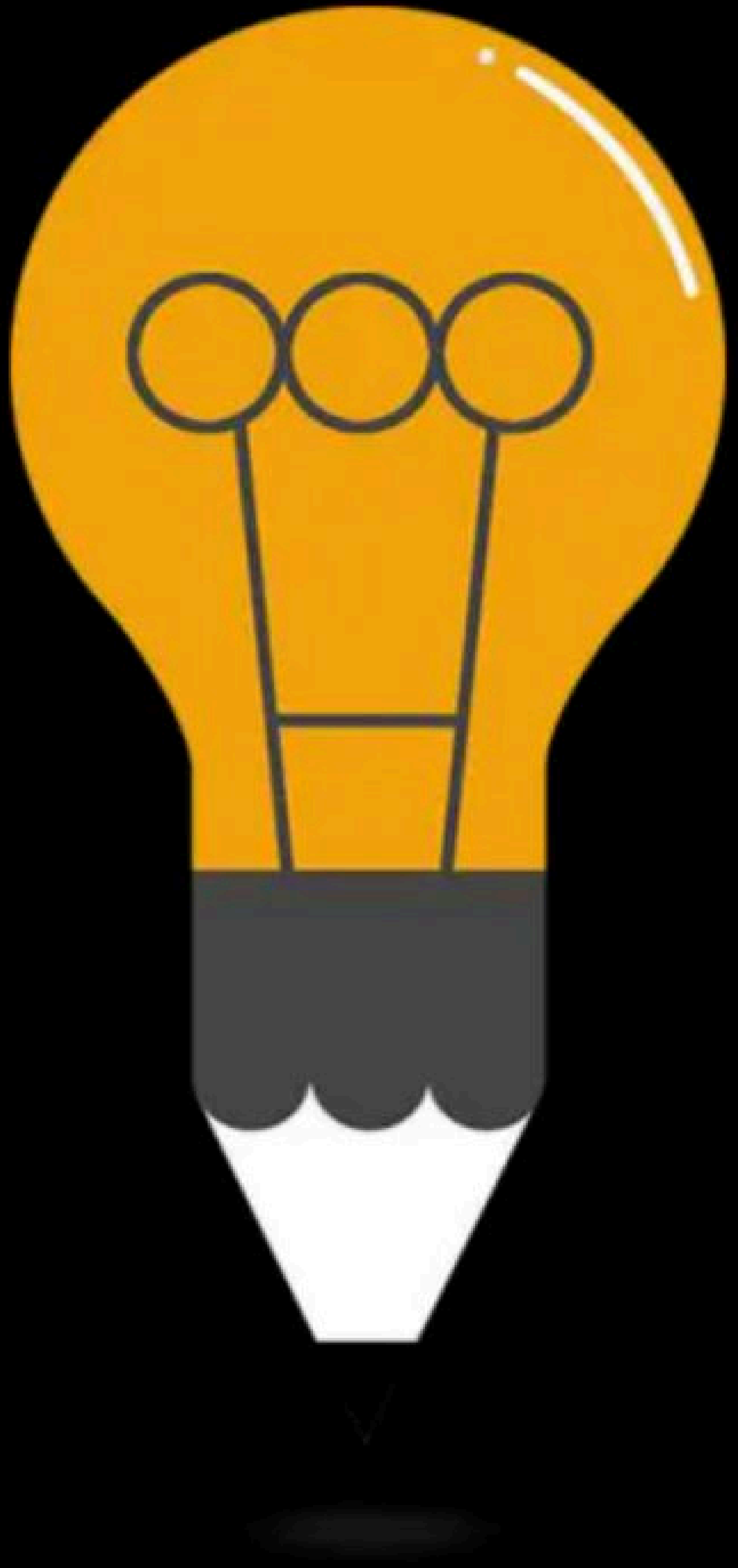
```
int fun (int x)
{
 return x/2;
}

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

x = ~~15~~ 7

x = 15

$\underbrace{\text{fun}(\underbrace{\boxed{\text{fun}(x)}}_{7})}_{3}$

3



DPP 5

By: Vishvadeep Gothi

Question

What is the output of the following programs-

```
int fun(int x,int y){
x=x + y;
y=x * y;
return (x, y);
}

int main(){
int x=4, y=8, z;
z=fun(x, y);
printf("%d", z);
}
```

Question

What is the output of the following programs-

```
void main(){
int fun(int);
int count=0, i;
for(i=1;i<1024; i*=2)
            count++;
printf("%d",fun(count));
int fun(int count) { return –count; }
}
```

Question

```
#include<stdio.h>
int fun(int);
int main(){
int a=fun(12);
printf("%d\n",--a);
return 0;
}
int fun (int x)
{ return x--; }
```

Question

What is the output of the following programs-

```
#include<stdio.h>
int fun(int n){
printf("%d", n--);
exit(0);
}

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

Question

What does the following function return when called for fun(511, 512)

```
int fun(int x,int y){
while(x!=y){
if(x>y) x=x-y;
else y=y-x;
}
return x;
}
```

Question

What is the output of the following programs-

```
#include<stdio.h>
int fun(int x)
{ return ++x; }

int main(){
int a=20;
a=fun(y=fun(y=fun(y)));
printf("%d", a);
return 0;
}
```

Question

What does the following function return when called for fun(1, 511)

```
int fun(int a, int b){
int z=1;
while(b>0){
if(b&1) z=z*a;
b=b>>1;
a=a*a;
}
return z;
}
```

Happy Learning.!