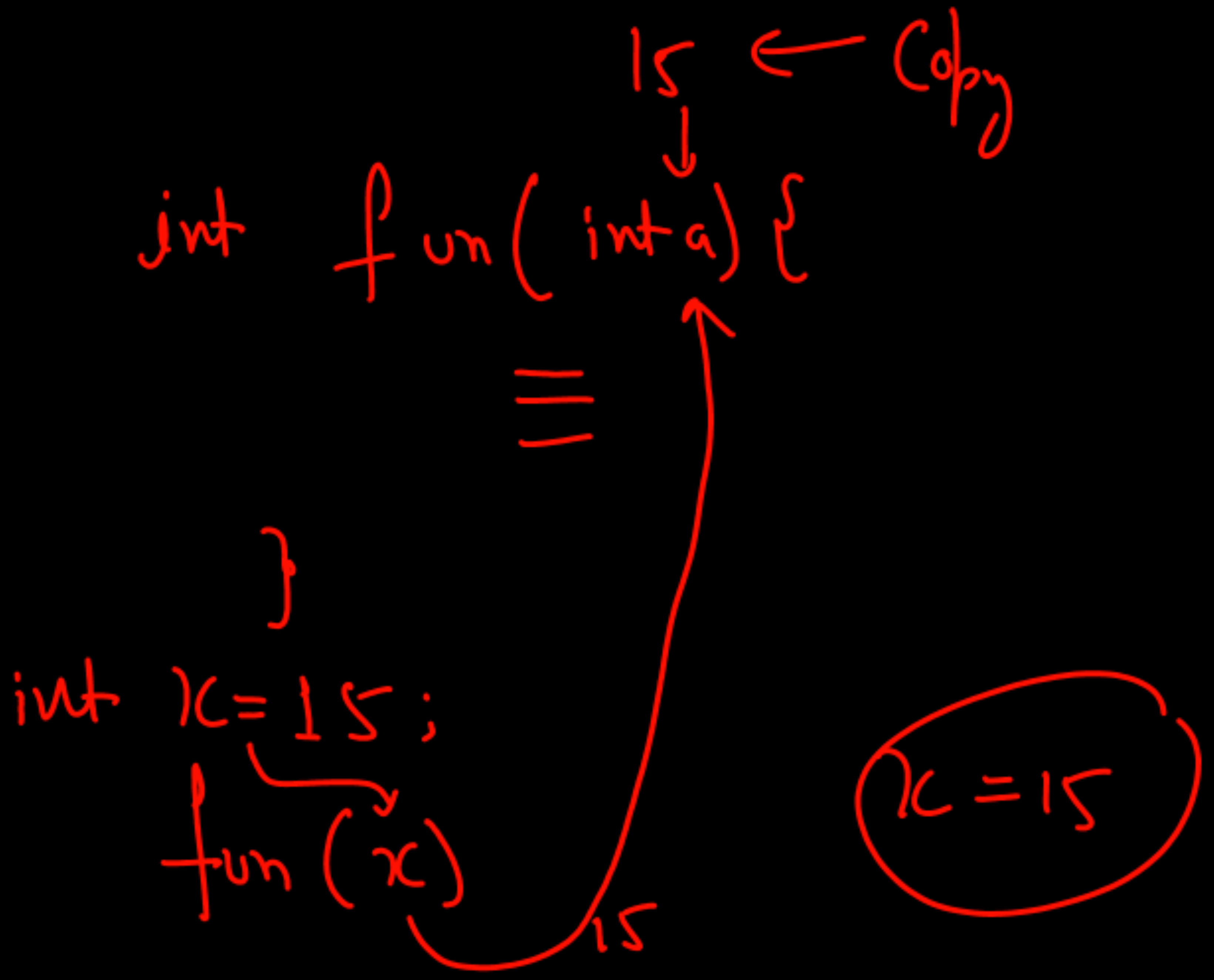
Lecture - 26

functions in C (Part - 04)

⇒ Programming in C

Call by Value :

↳ Actual value passed during function call



```
int fun( int a) {
      ≡
}
int x = 15 ;
    fun(x)
```

```
void fun1( int a) {
    a++ ; ✓
    printf(" %d\n", a) ;
}

int main ( ) {
    int a= 10 ;
    fun1 ( a);
 ⇒ printf("%d \n", a);
    return 0;
}
```

10

11

copy

10

fun1 (a); ← Call → fun1(a)

10 ← Call by Value

10

fun 1

↳ a= ~~10~~ 11

main

↳ $\boxed{a= 10}$

Pointers:

↳ Are the variables that stores the address of other variables.

```
int main() {

    int x = 24;
    int* p = &x;

    int y = 50;
    int* q = &y;
```

p is a pointer variable that is holding the address of x.

q is also a pointer

&x → $\text{int } p = \&x = 100$

$\text{int } q = \&y = 200$ ← &y

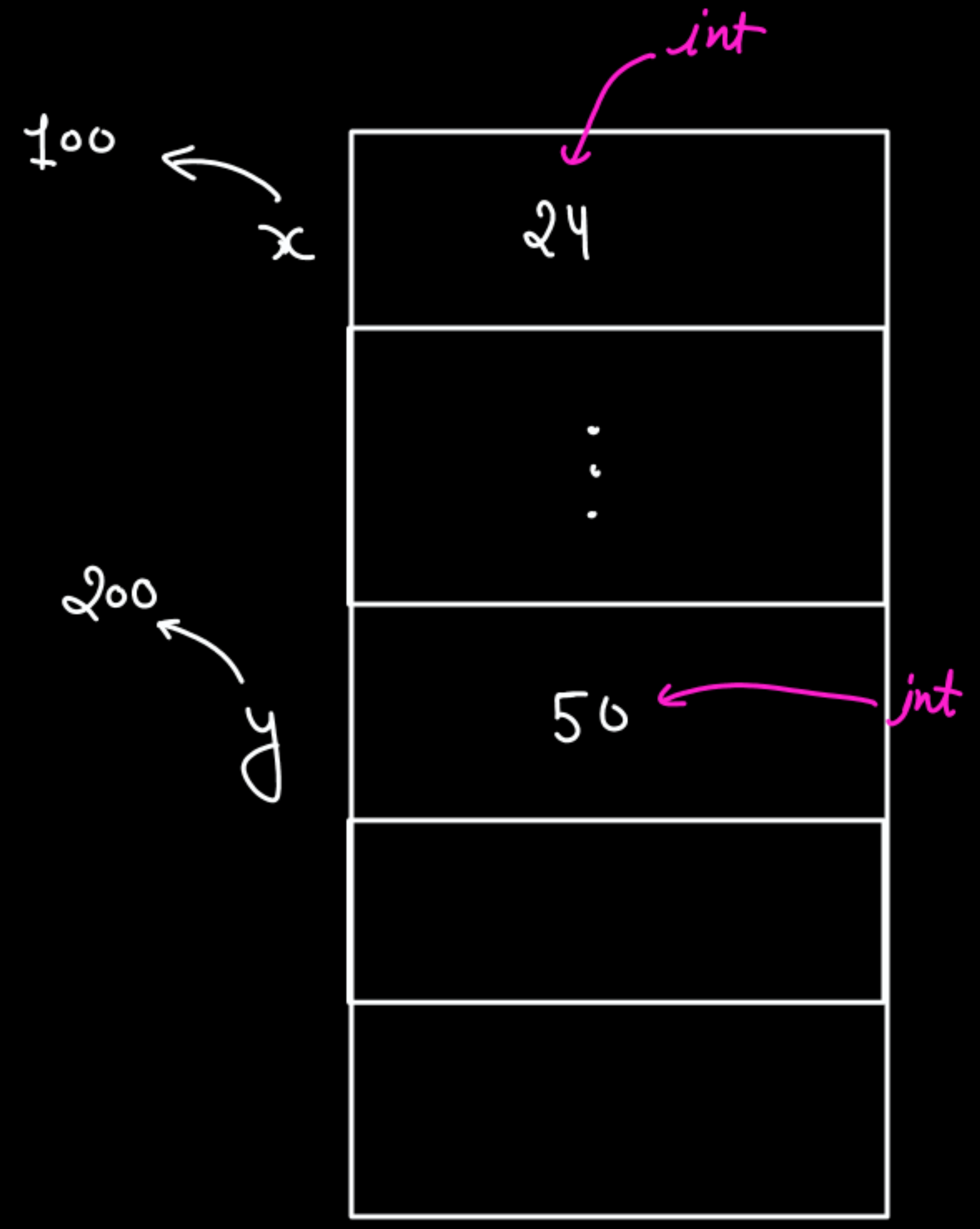
* ← Dereferentiation operator

```
printf("%d", p);
```

← Garbage Value

%d → int

p → pointer



```
int main() {
    int x = 20;
    int* m = &x;
    printf("%p", m);
    printf("%d", *m)
    return 0;
}
```

Addr of x = 501

Pointer

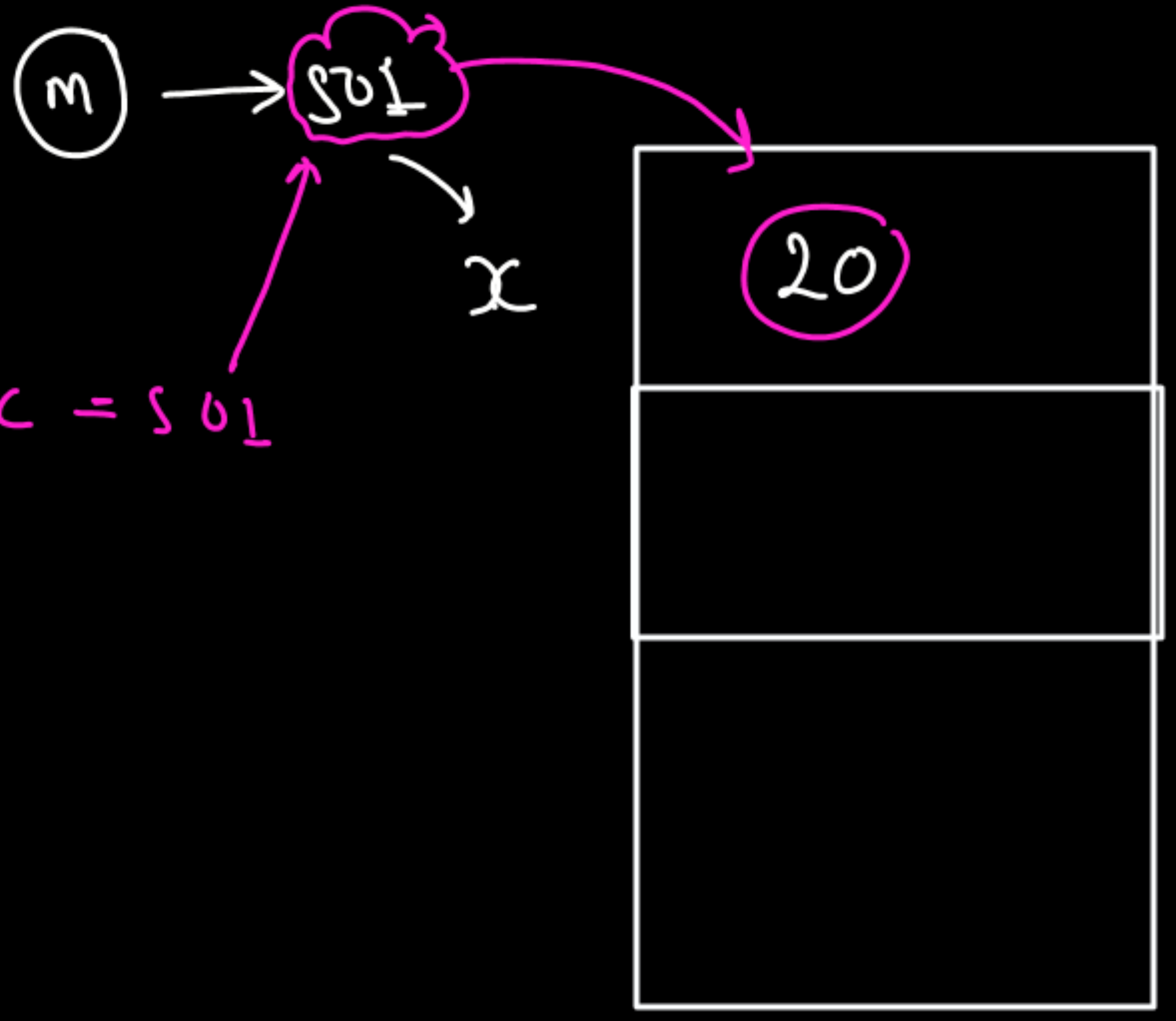
501

int

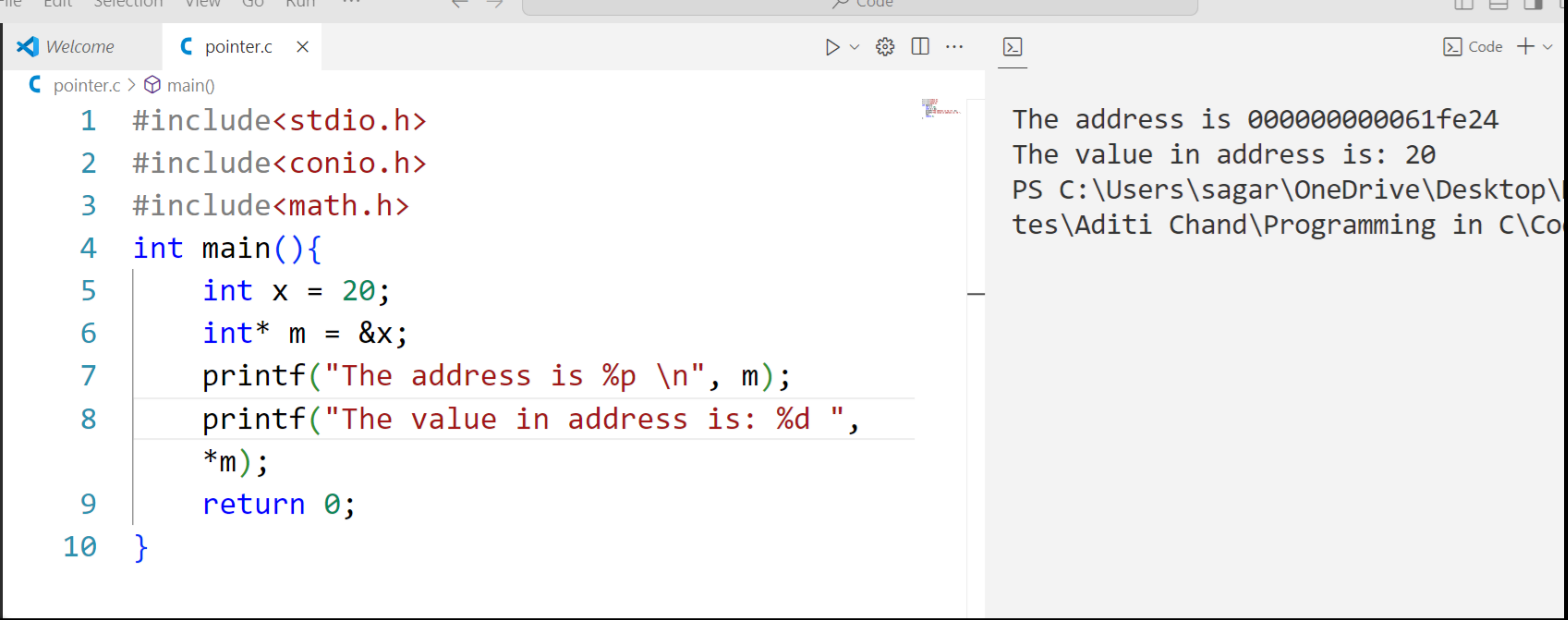
20

de-reference करो m को

↳ Getting inside the address of m
↓
20 ← int



* ← de-reference



```c
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main(){
    int x = 20;
    int* m = &x;
    printf("The address is %p \n", m);
    printf("The value in address is: %d ", *m);
    return 0;
}
```

```
The address is 000000000061fe24
The value in address is: 20
PS C:\Users\sagar\OneDrive\Desktop\
tes\Aditi Chand\Programming in C\Co
```

```
Void fun ( int* a) {
    *a = *a + 10;
    printf ("%d\n", *a)
}

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

address

pointer

30

$*a = *a + 10$

$20 + 10$

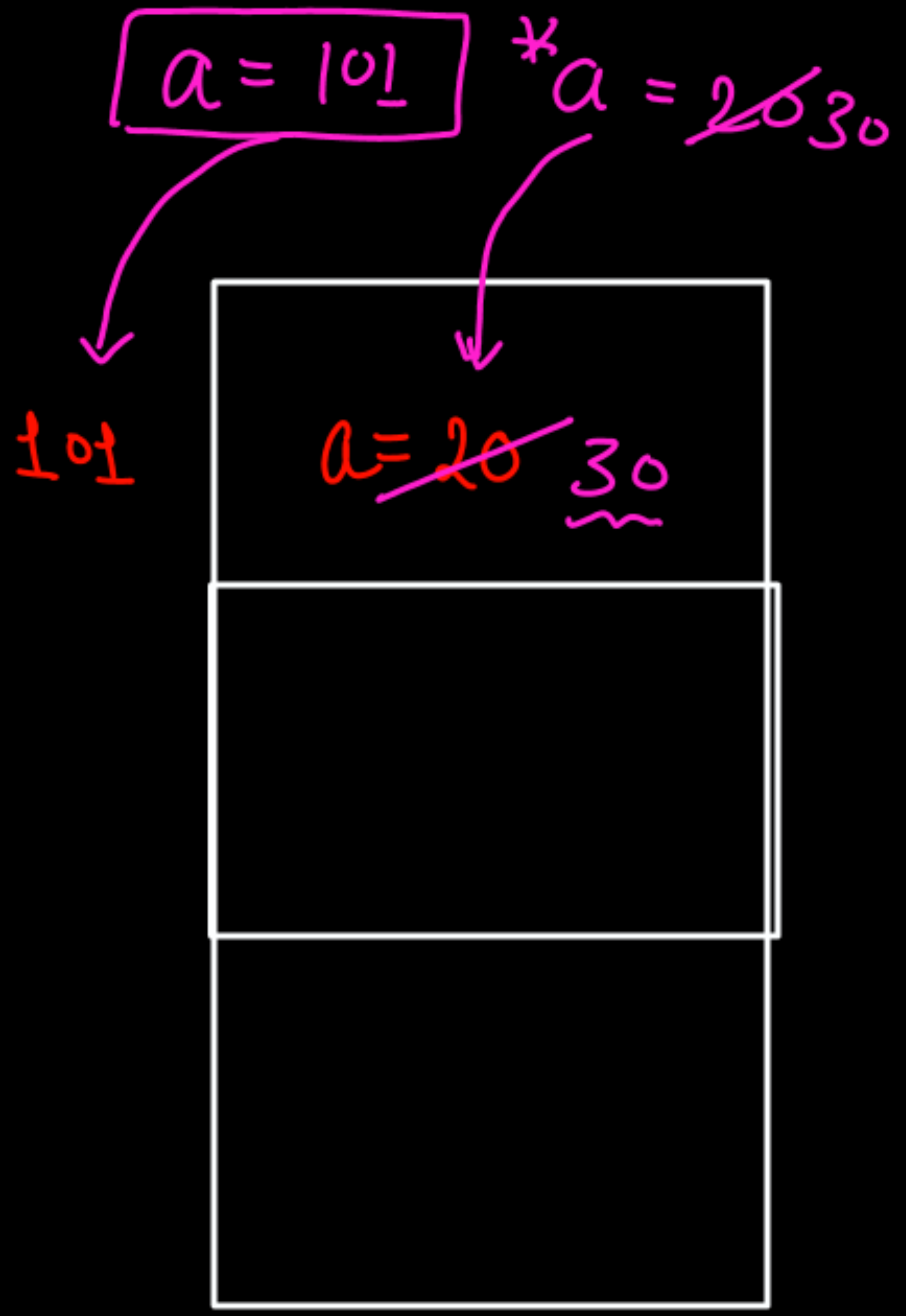
$\boxed{*a = 30}$

a → Value = 20

→ address = 101

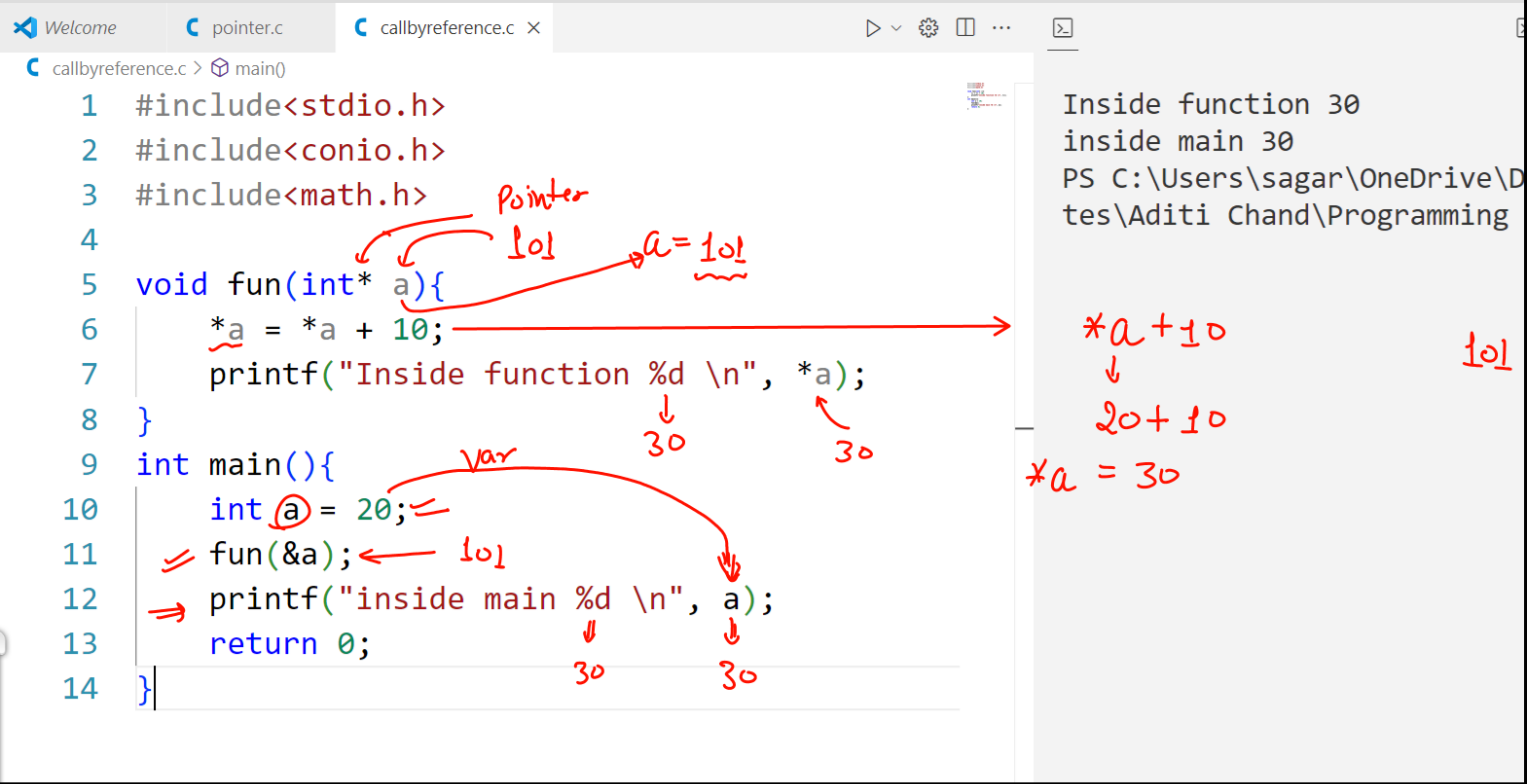
101

30



O/P

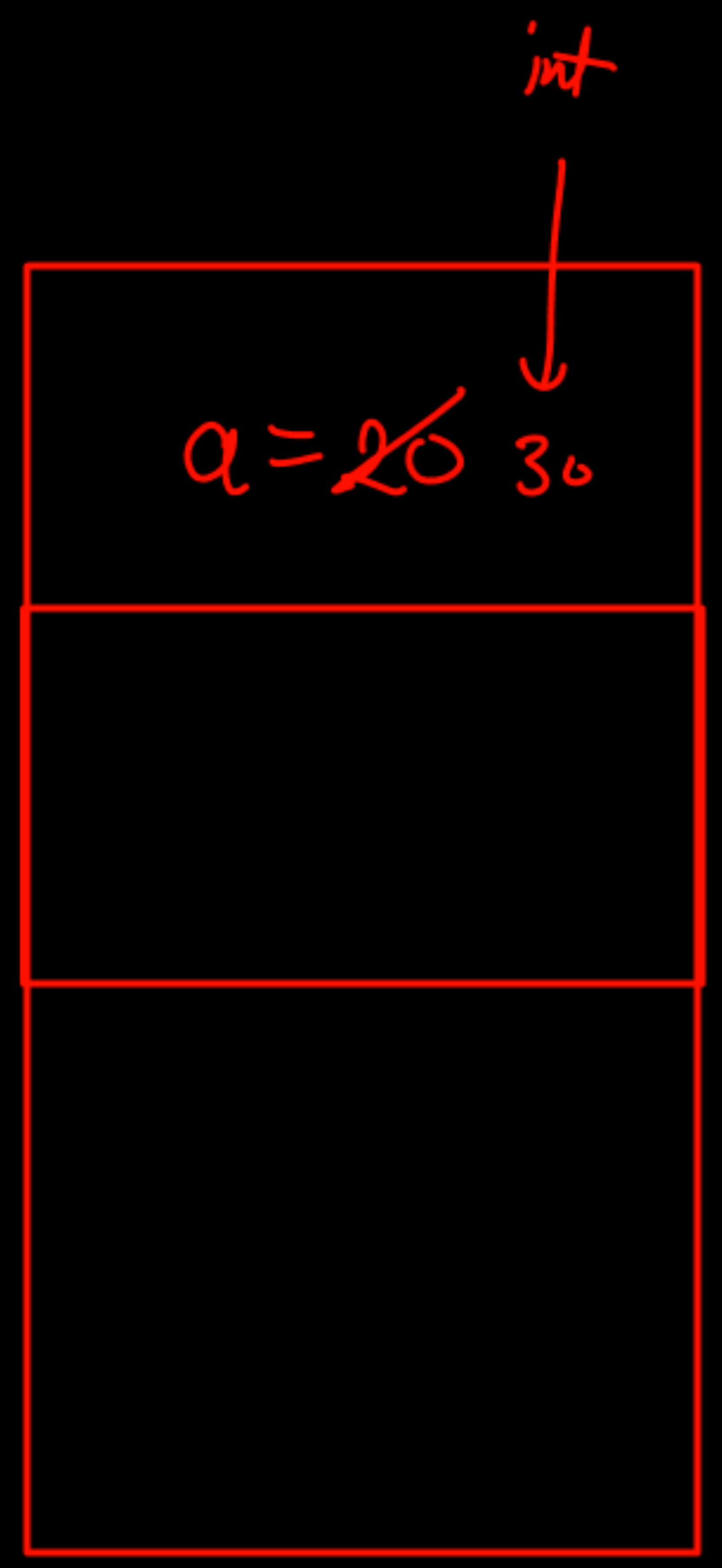
⇒ 30



```c
#include<stdio.h>
#include<conio.h>
#include<math.h>

void fun(int* a){
    *a = *a + 10;
    printf("Inside function %d \n", *a);
}
int main(){
    int a = 20;
    fun(&a);
    printf("inside main %d \n", a);
    return 0;
}
```

```
Inside function 30
inside main 30
PS C:\Users\sagar\OneDrive\D
tes\Aditi Chand\Programming
```



fun → a is a pointer ✓
main → a is a variable ✓] Both are different

Tomorrow

⇒ global variable

⇒ Recursion

⇓

Storage Classes