unacademy

Arrays & Pointers - Part X

Comprehensive Course on C- Programming

Pankaj Sharma • Lesson 28 • June 6, 2024

unacademy

CS & IT Engineering

C Programming

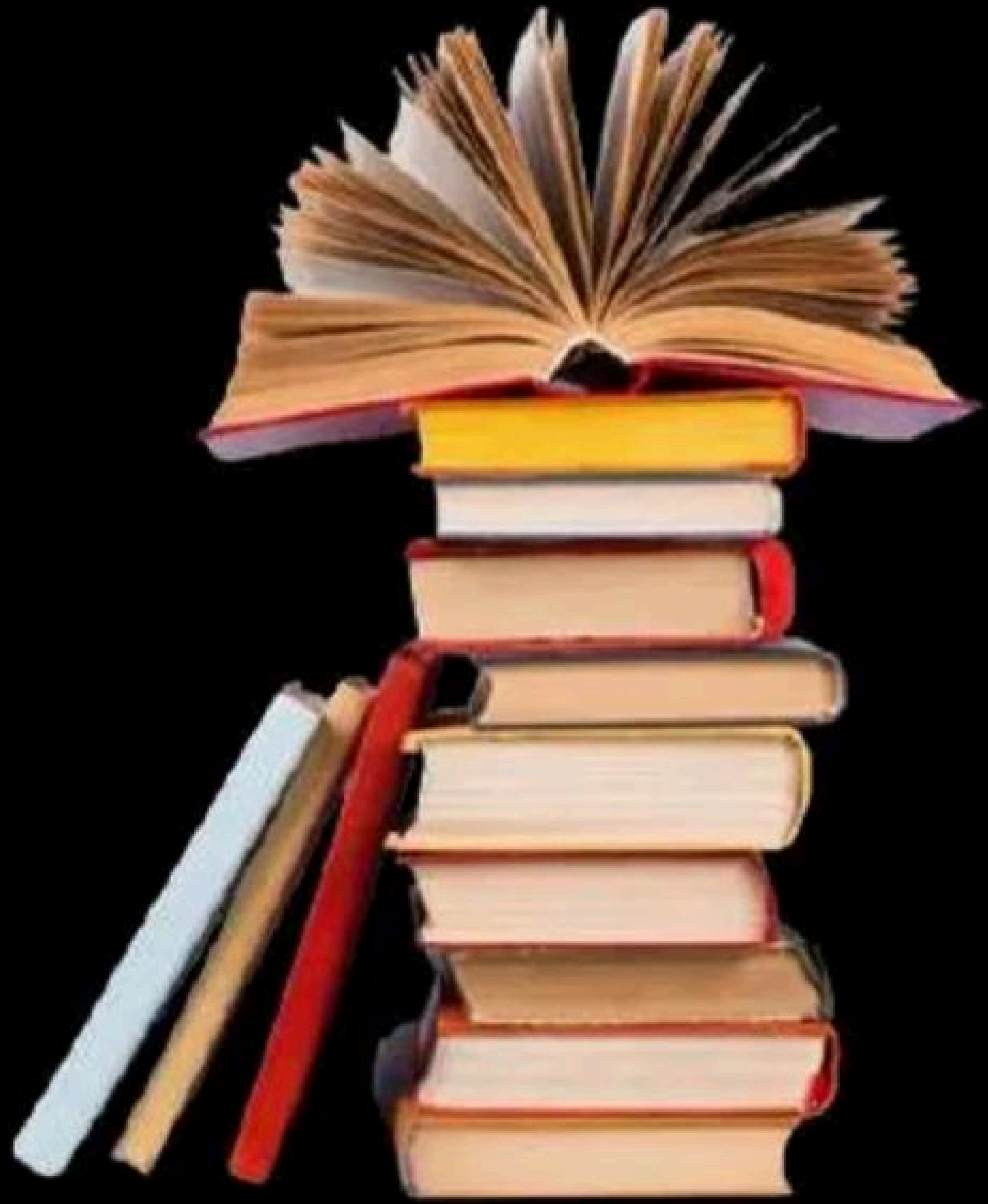
Arrays & Pointers-X

Lecture Number- 28

By- Pankaj Sir

Topics

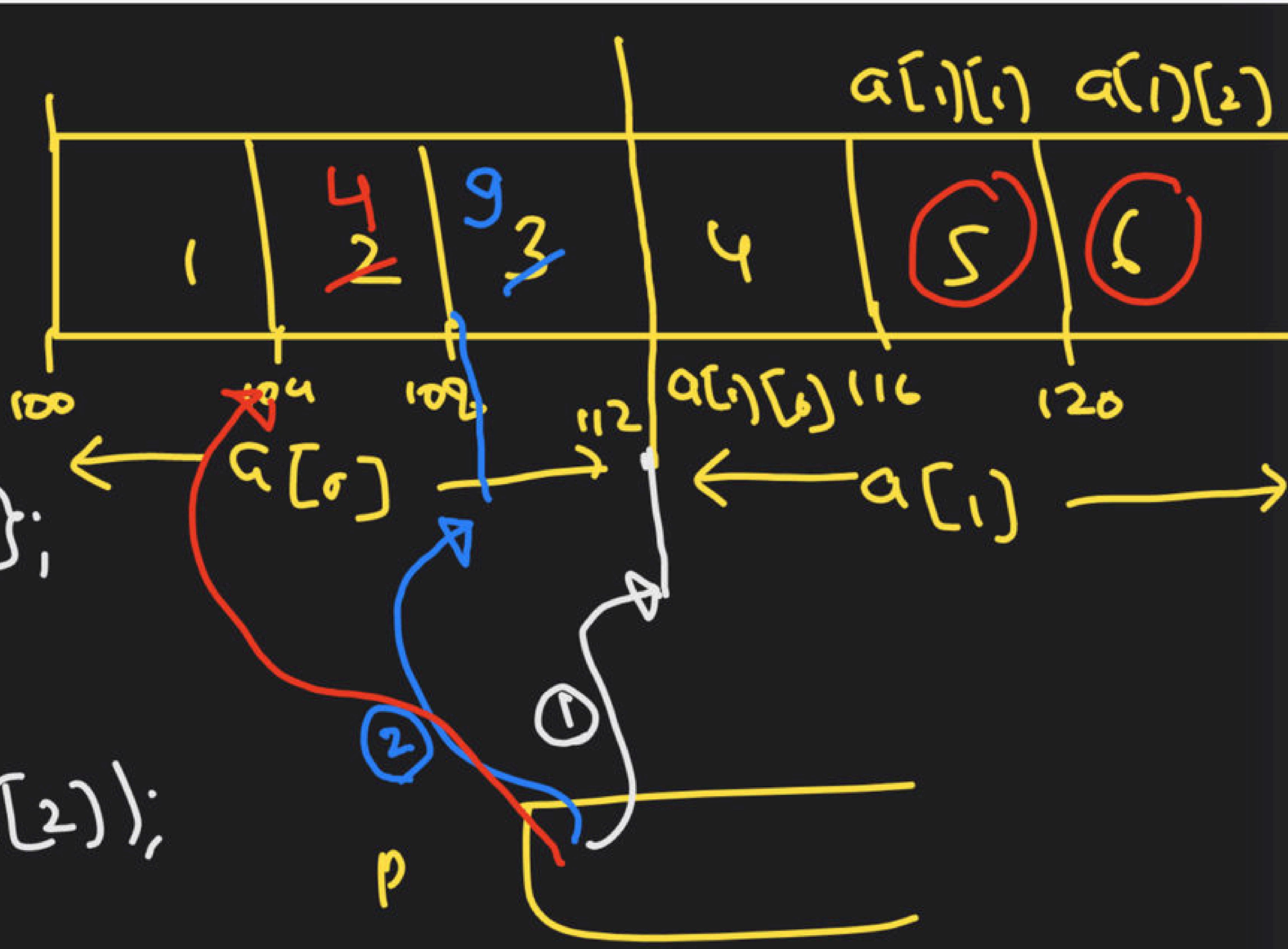
to be covered



1. Arrays & Pointers -X

unacademy

```
Q  void f(int*);
   void main(){
   int a[2][3] = {1,2,3,4,5,6};
   f(a[1]);          &a[1][0]
   pf("%d %d", a[1][1], a[1][2]);
            }    5 6
   void f(int *p){ p--;
          *p = *p * *p;   ⇒  *p = 3 x 3
          p--;  ✓
          *p = *p * *p; }  →  *p = 2 x 2
```



unacademy

Q

```
int a[4][5] = {{1,2,3,4,5},
               {6,7,8,9,10},
               {11,12,13,14,15},
               {16,17,18,19,20}};

pf("%d", *(*(a + **a+2)+3));
```

$$**a = *\cancel{*}\cancel{\&}a[0]$$

$$= *a[0]$$

$$= \cancel{*}\cancel{\&}a[0][0]$$

$$= a[0][0]$$

↓

$$*(*(a+1+2)+3)$$

$$*(*(a+3)+3)$$

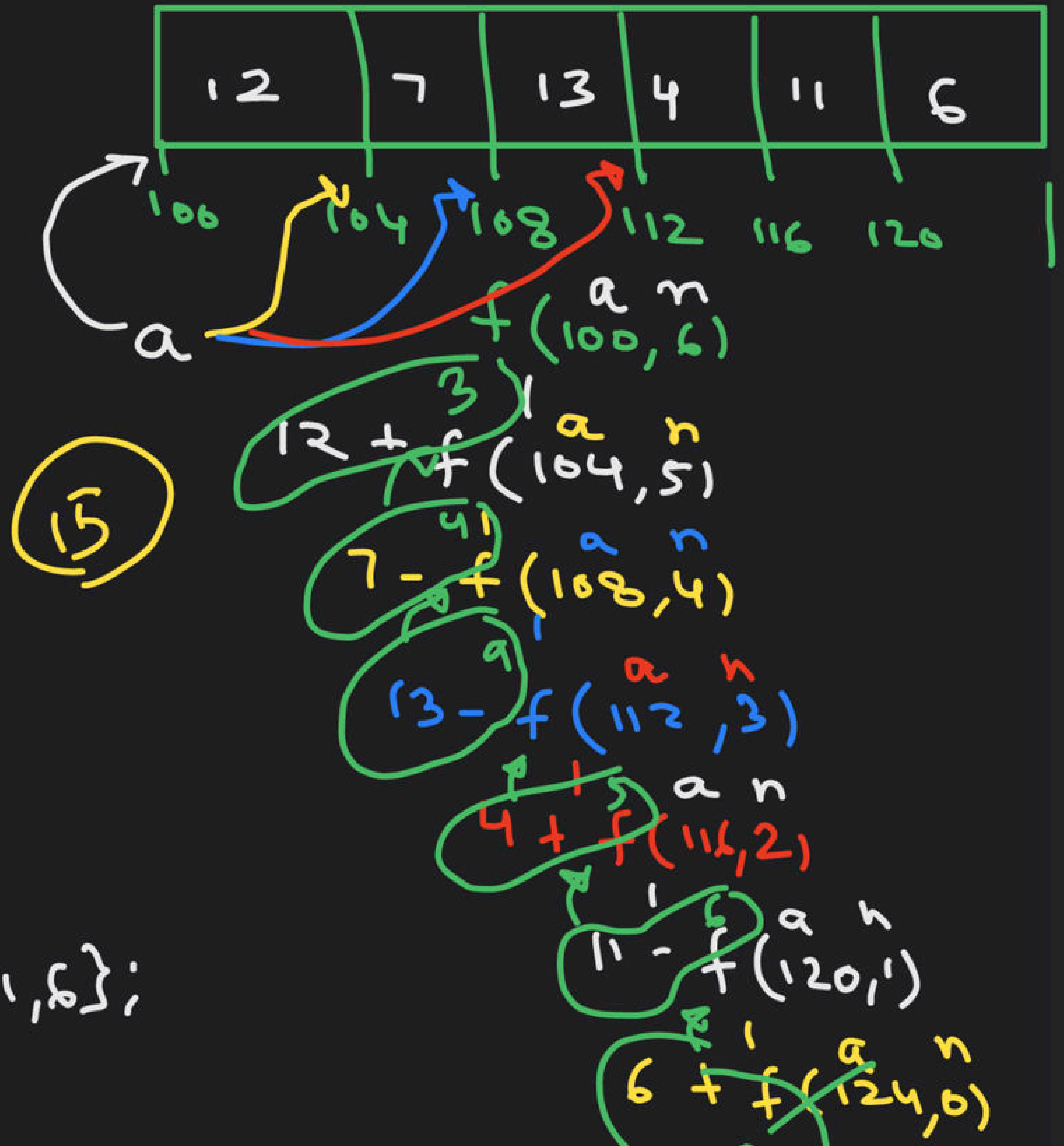
$$*(a[3]+3)$$

$$a[3][3]$$

$$\boxed{19}$$

unacademy

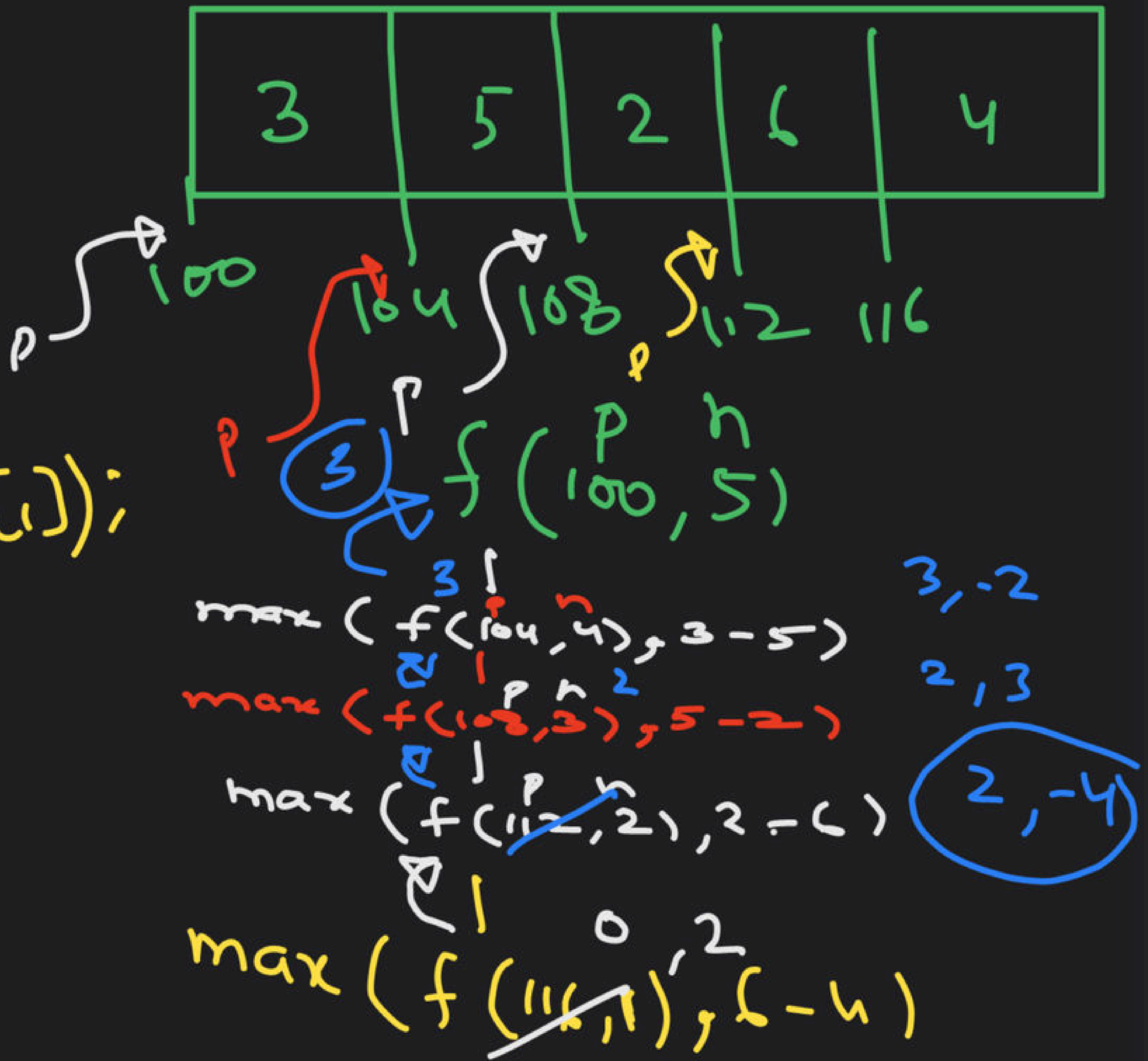
```
Q   int f(int *a, int n){
      if(n<=0) return 0;
      else if(*a % 2 == 0)
     return *a + f(a+1, n-1);
       else
      return *a - f(a+1, n-1);
                              }
  void main() {
     int a[] = {12,7,13,4,11,6};
      pf("%d", f(a,6)); }
```



Q

```
int f(int *p, int n){
    if(n<=1) return 0;
    else
  return max(f(p+1,n-1),p[0]-p[1]);
   }
void main(){
  int a[]={3,5,2,6,4};
  pf("%d",f(a,5));
   }
```

$\boxed{3}$



3 ← f(100, 5)

max(f(104, 4), 3-5)   3, -2

max(f(108, 3), 5-2)   2, 3

max(f(112, 2), 2-6)   2, -4

max(f(116, 1), 6-4)   0, 2

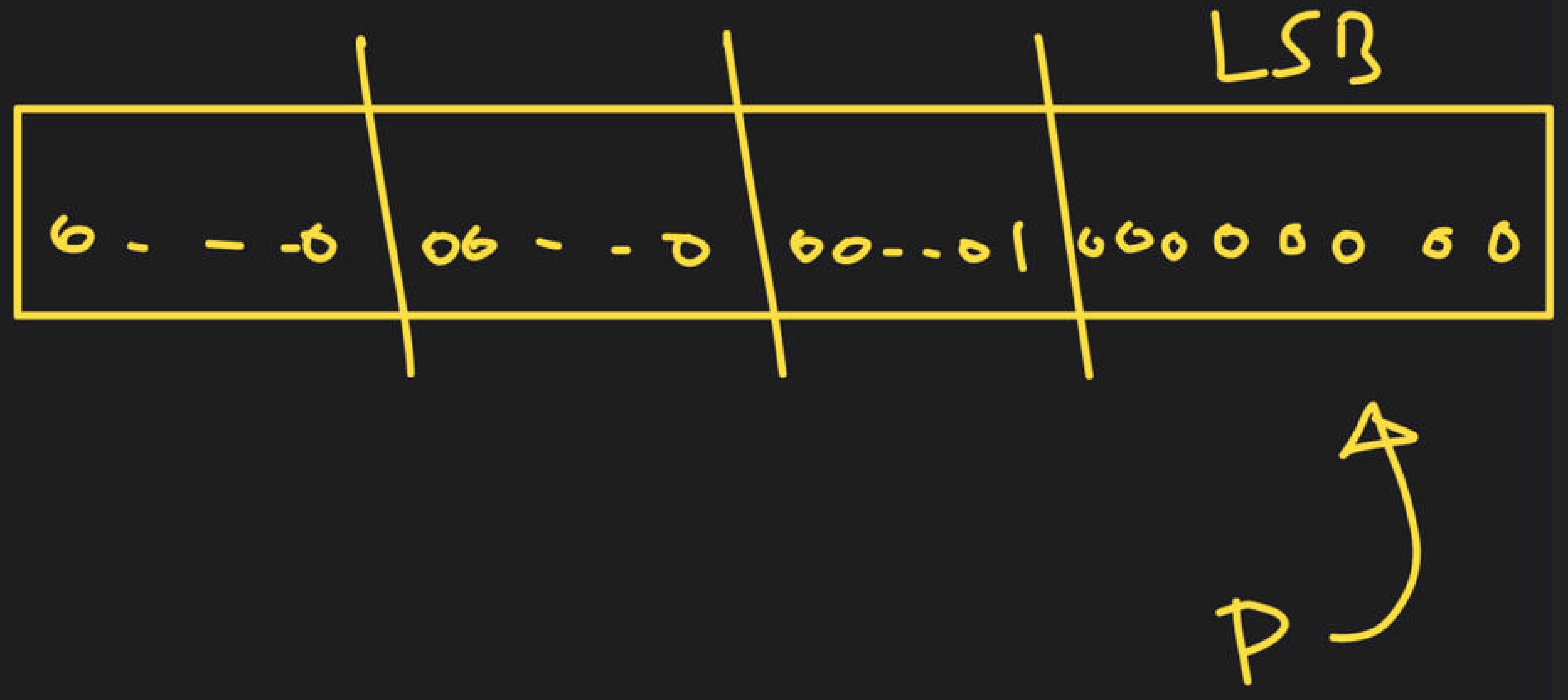
unacademy

Q

```
int a=256;
char *p;
p = (char*)&a;
pf("%d",*p);
```

0

Little Endane



LSB

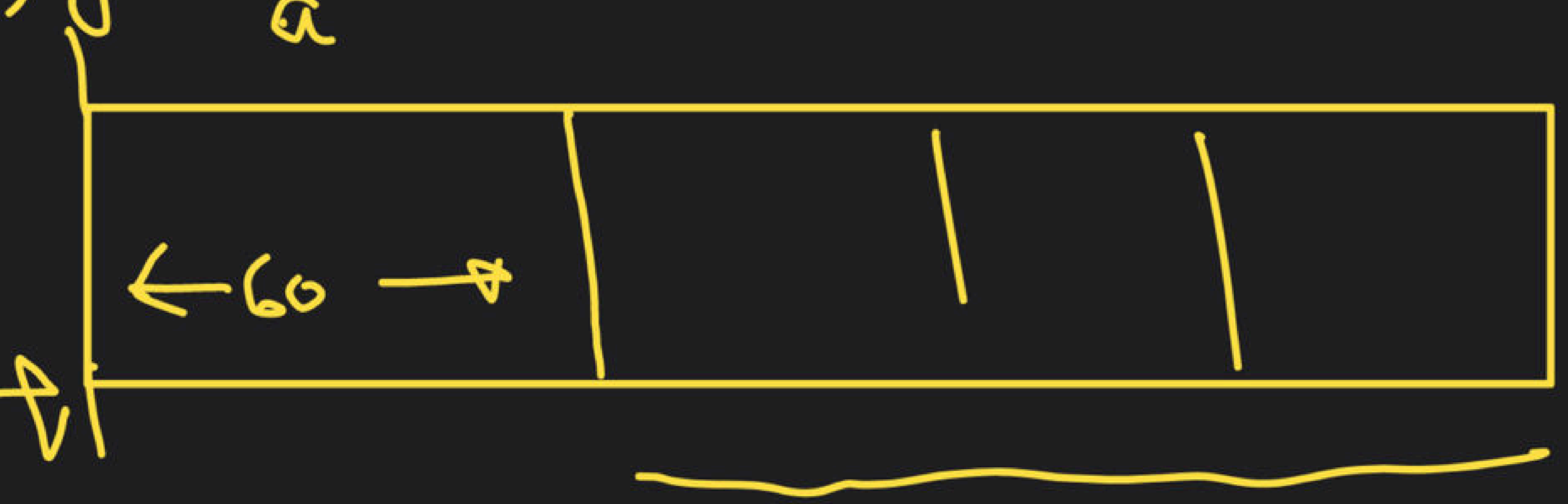
0 - - -0	00 - - -0	00- -01	00000000

unacademy

LSB / MSB

```
char a = 60;
int *p;
    p = (int*)&a;
pf("%d", *p);
```

big

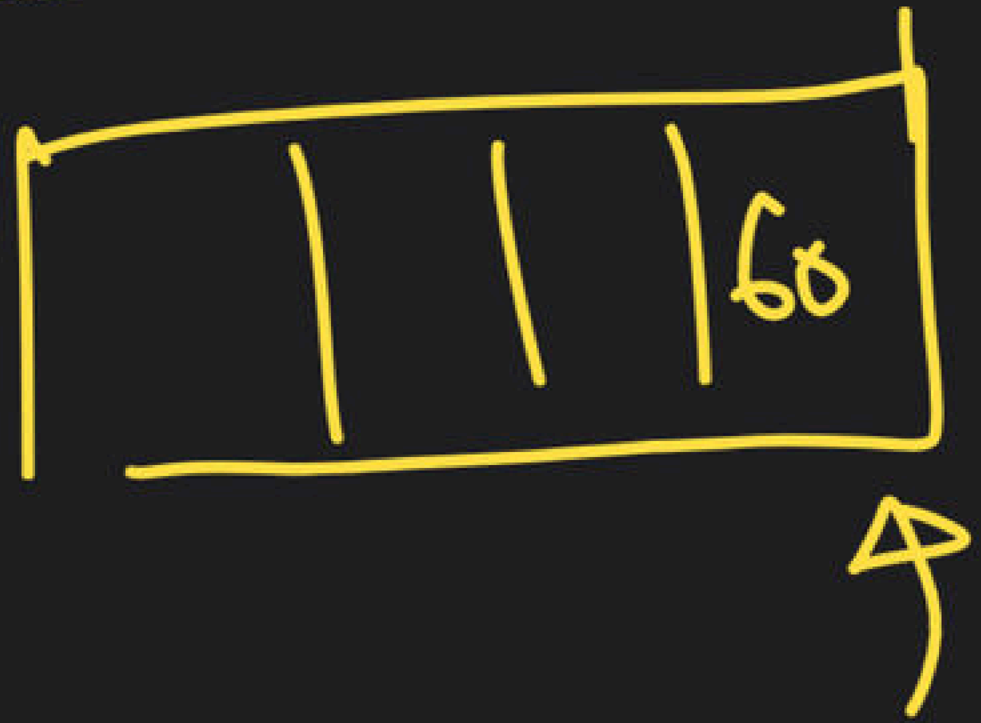


4byte

p

Garbage

little End.



Doubt ?

unacademy

2

int *p; → 4 byte

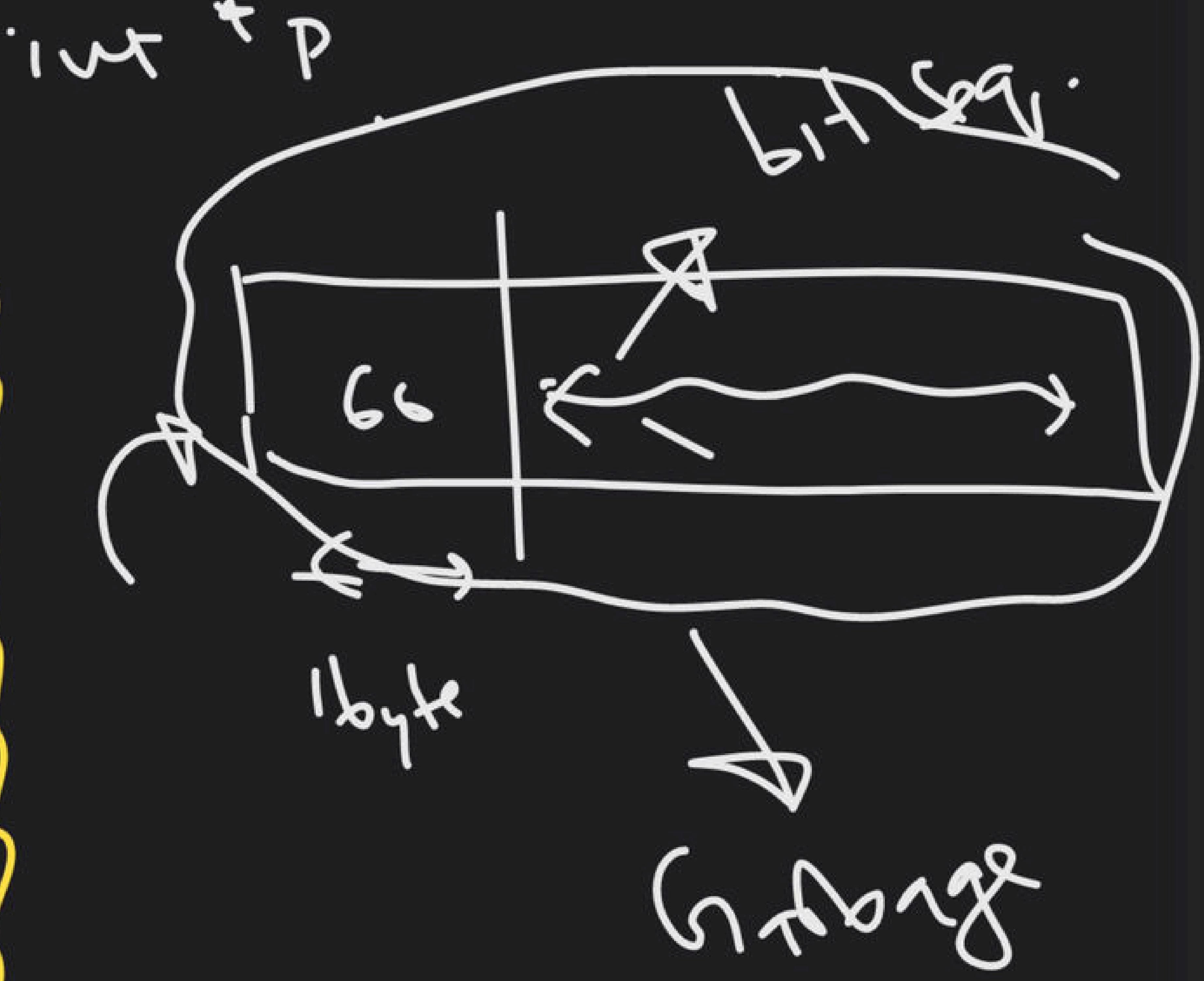
≡

P=p+2; (8 bytes)

⇓

moving 2 location in forward direction

Neena



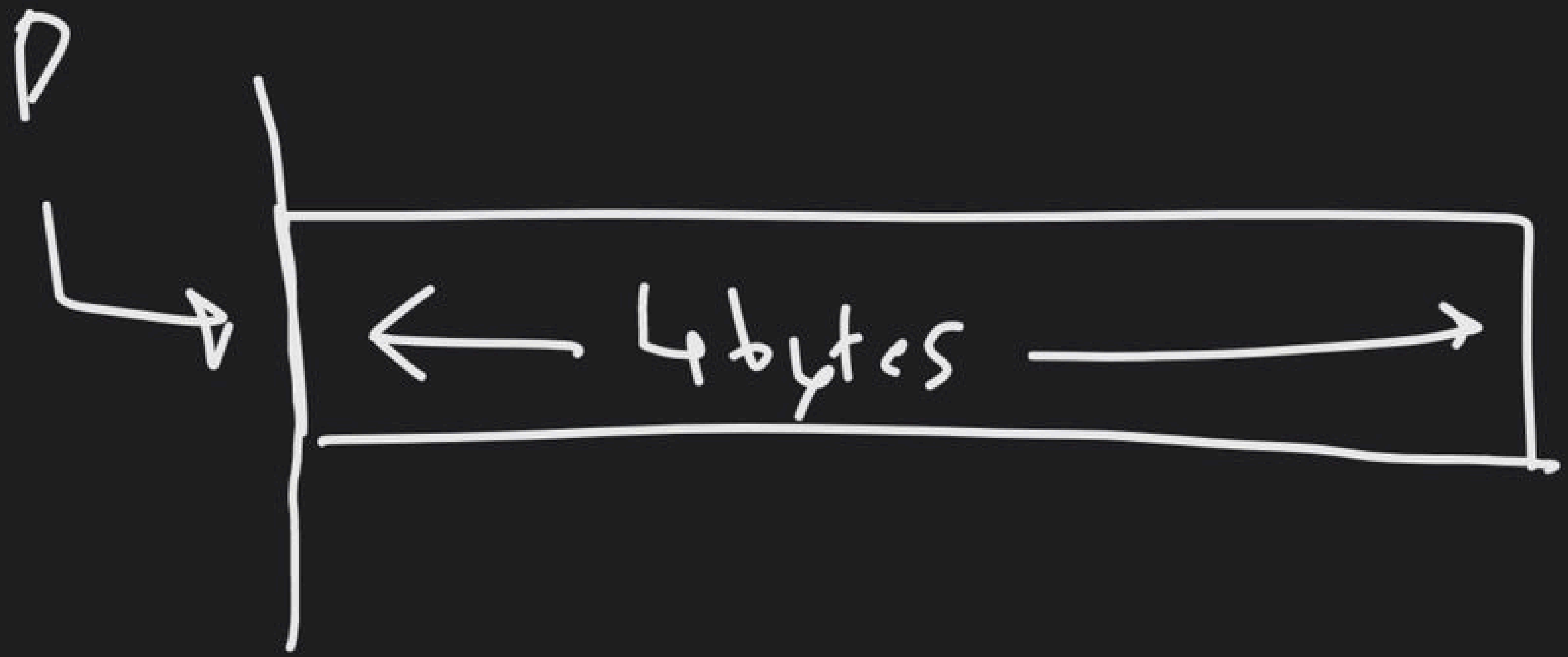
unacademy

$\boxed{\text{char}}$ *p;

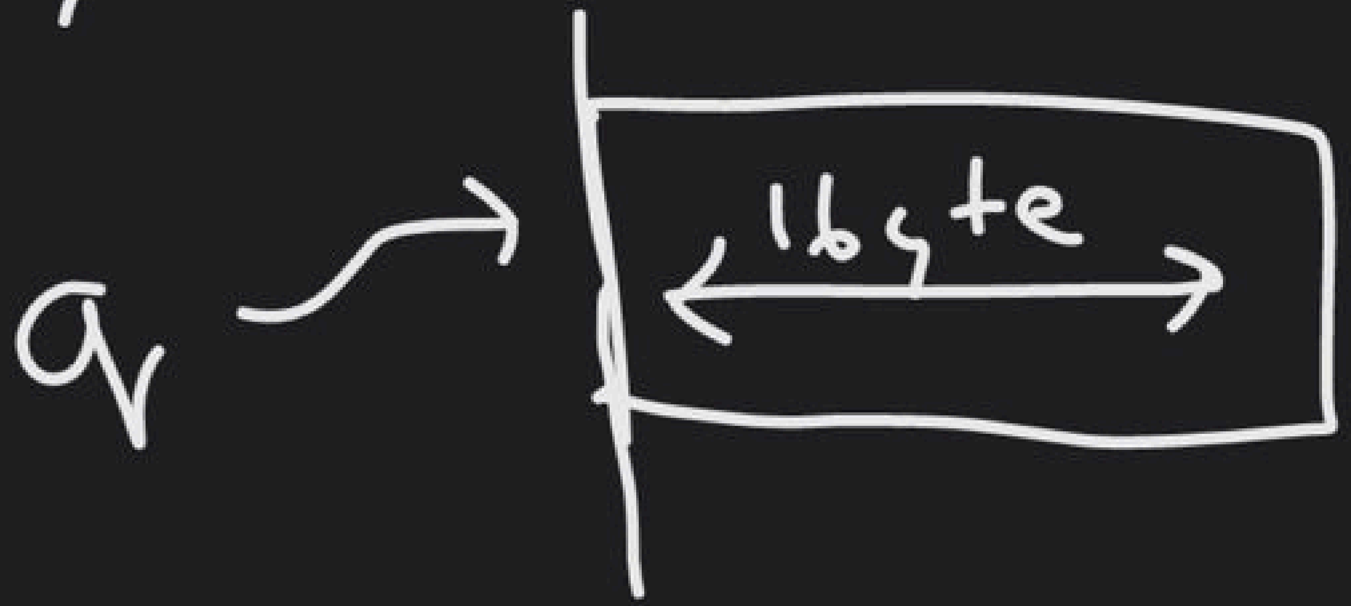
p = p+2; ⇒ (2 byte ⇒)

unacademy

int *p;



char *q;



unacademy

void pointer

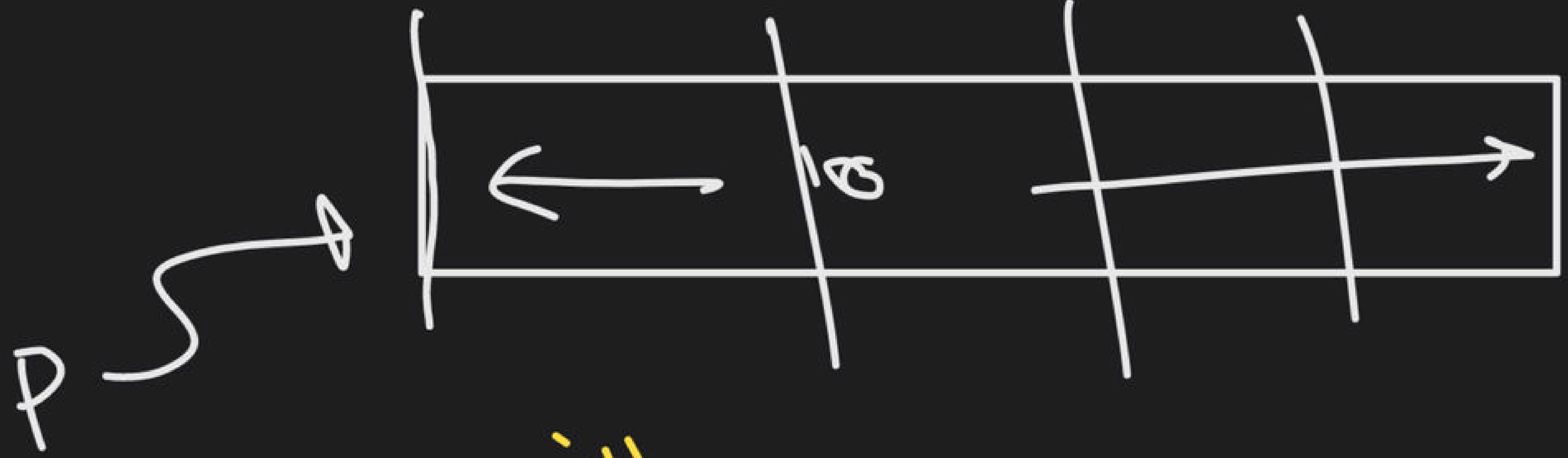
void *p;   int *p;   char *p;

```
void *p;

int a = 10;

float b = 4.32;
```

$\boxed{p = \&a;}$ → compiler will shout or not



```
pf("%d", *p);
```

→ Don't directly dereference any void pointer

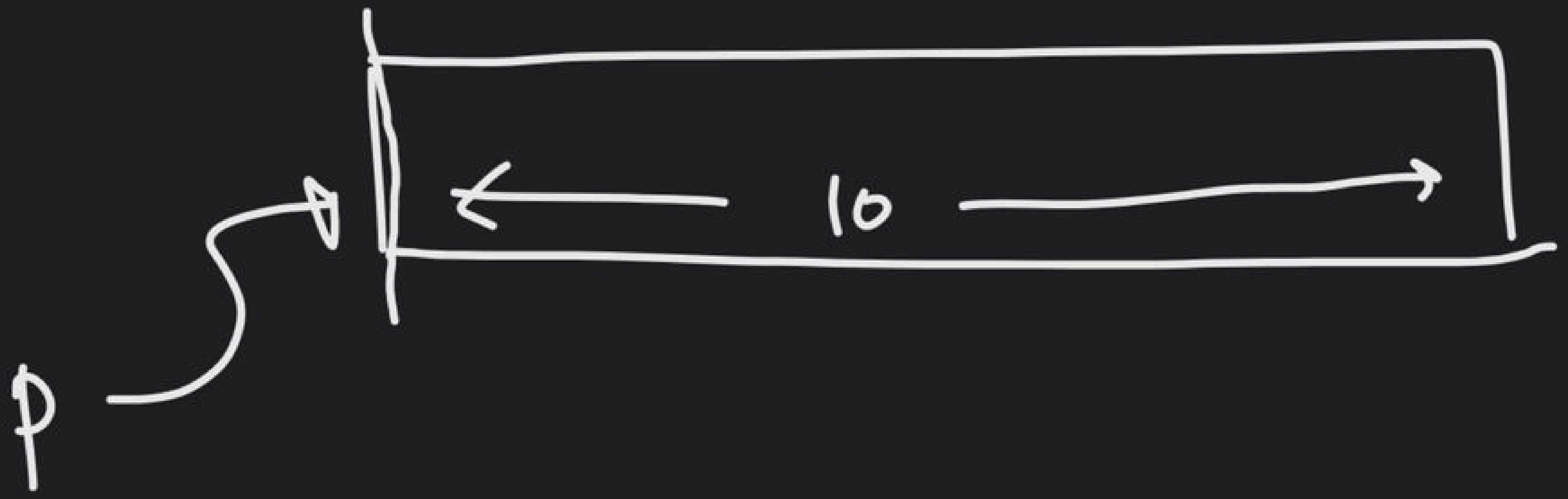
```
void *p;
int a=10;
p=&a;
```

✓

pf("%d",*p); ⇒ Error

↓ Solution

pf("%d", *(int*)p) ✓



P ⇒ (int*)p

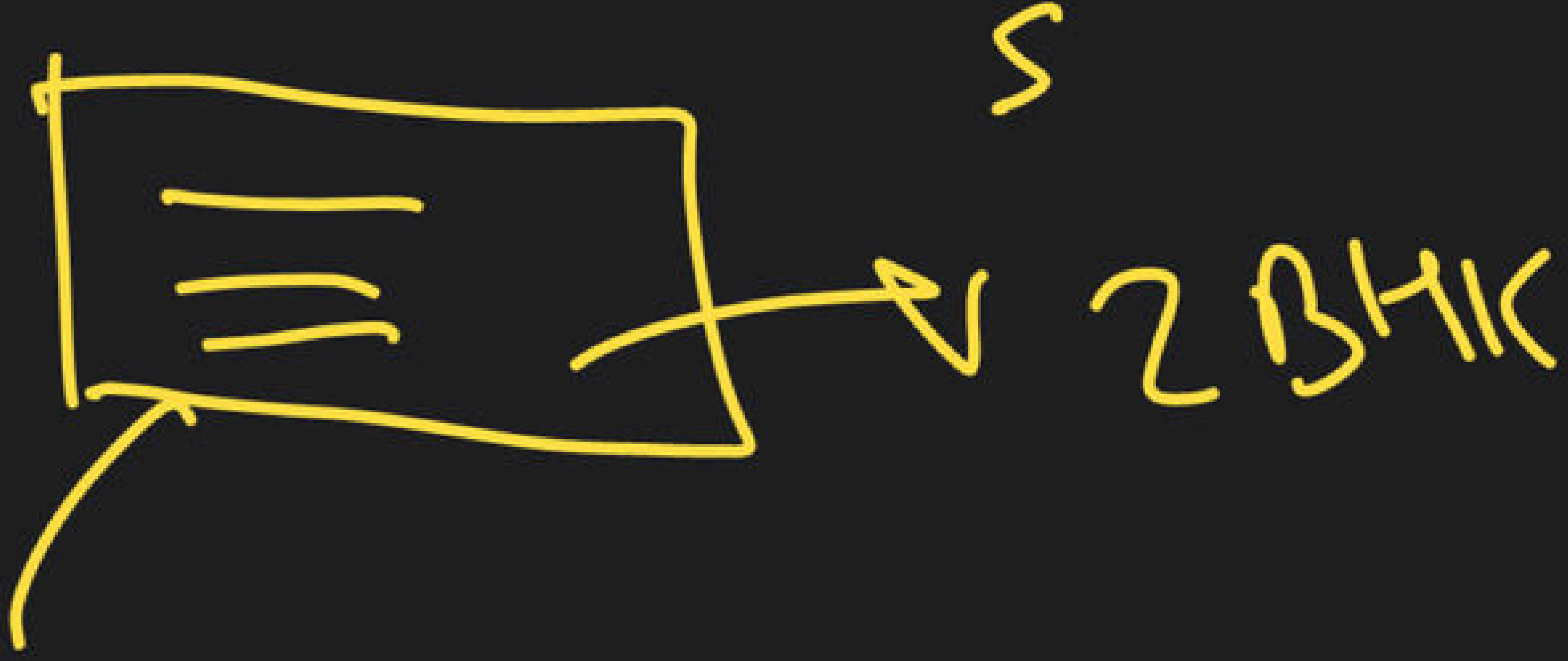
(int*) ⟶ typecasting

⑩

unacademy

① we can not derefrence a void pointer directly

② First typecast then only derfrence it.



unacademy

int *p;

p = p+2

↓

2 * sizeof(int)

char *p;

p = p+2

↓

2 * sizeof(char)

unacademy

```
void *p;

p = p+2;
```

Do not perform any arithmatic operation on void pointer

Wild pointer

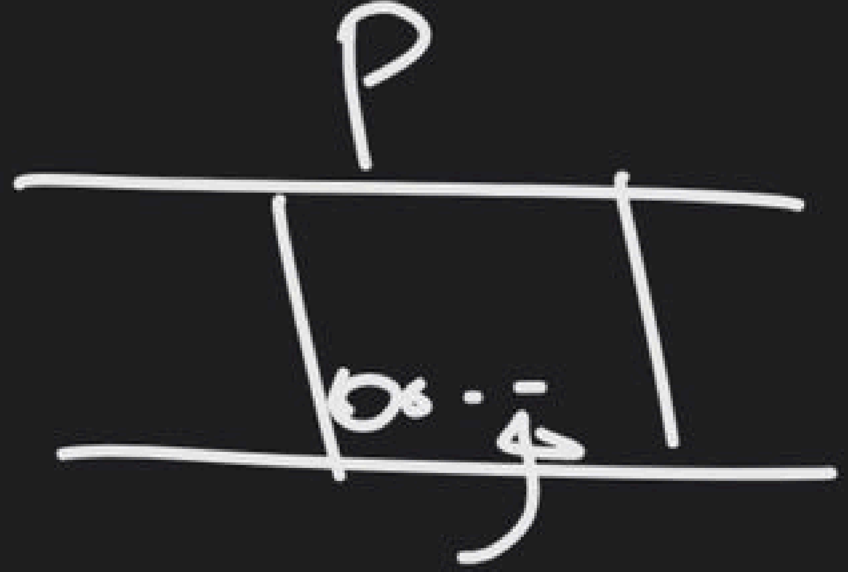
→ uninitialized pointe

___

```
void main(){
    int a;  Garbage
    pf("%d",a);
}
```

128
-96
30000
0

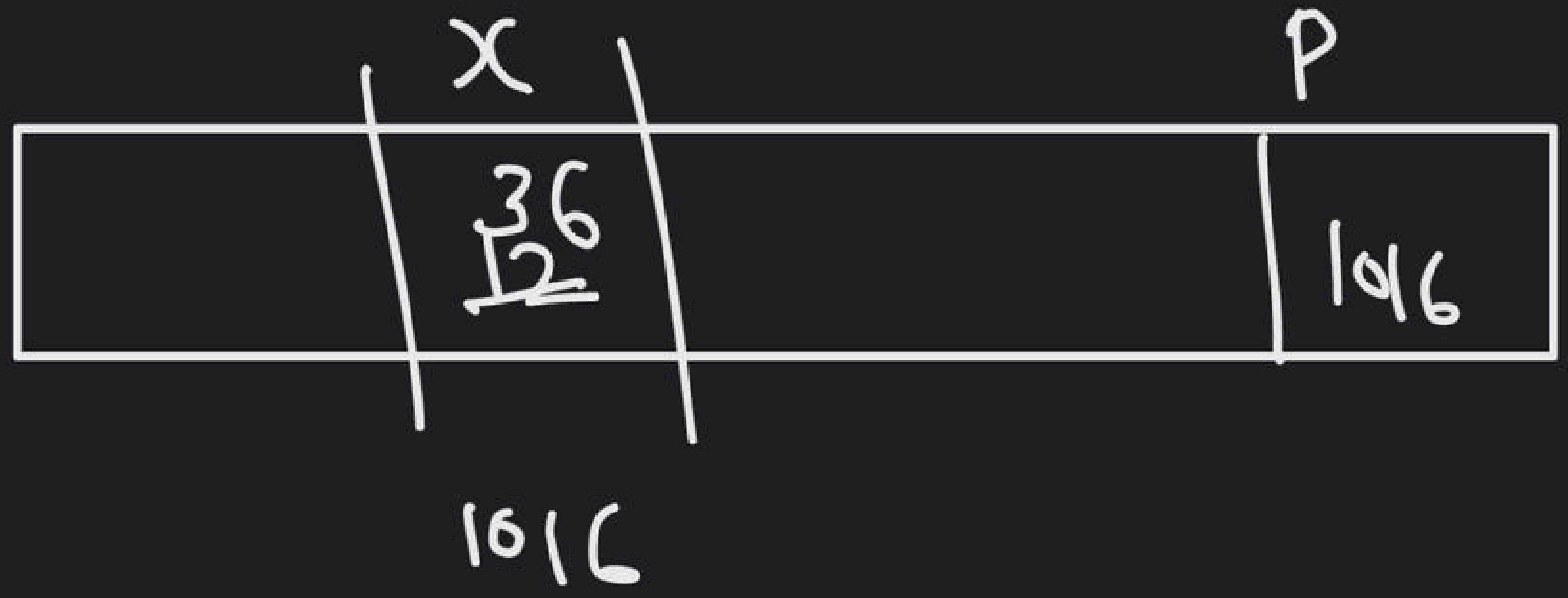
```
void main(){   Garbage
    int *p;
    =
}
```



unacademy

negligible

```
void main(){
   int *p;
   int x=12;
   *p = 36;
}
```



2nd problem!

Elevate batch

unacademy

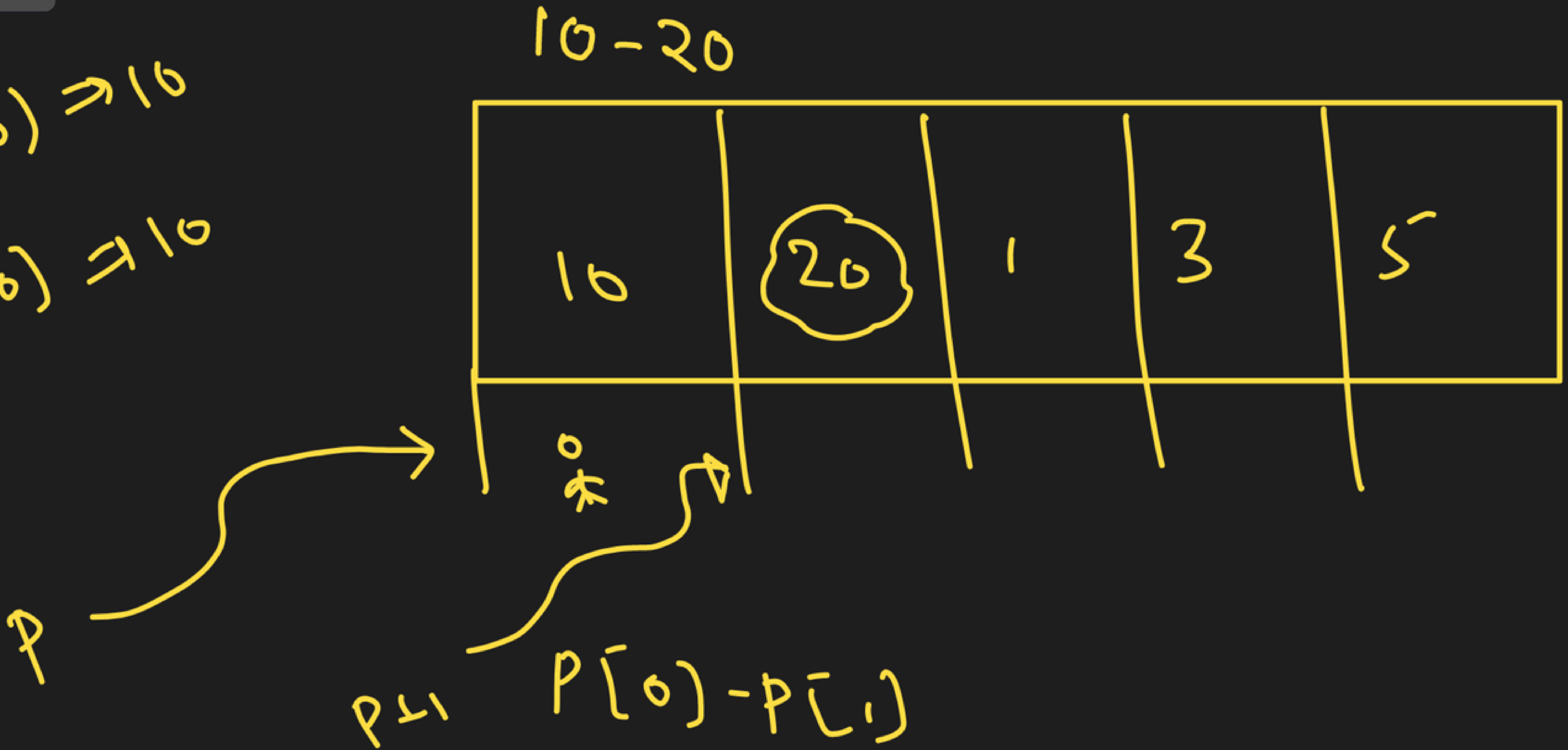
Sayan → accuracy → ✓✓ →

Sayan → Speed

- void pointer
- wild pointer
- dangling pointer
- NULL pointer
- Dynamic memory allocation

$*(p+0) \Rightarrow 10$

$p[0] \Rightarrow 10$

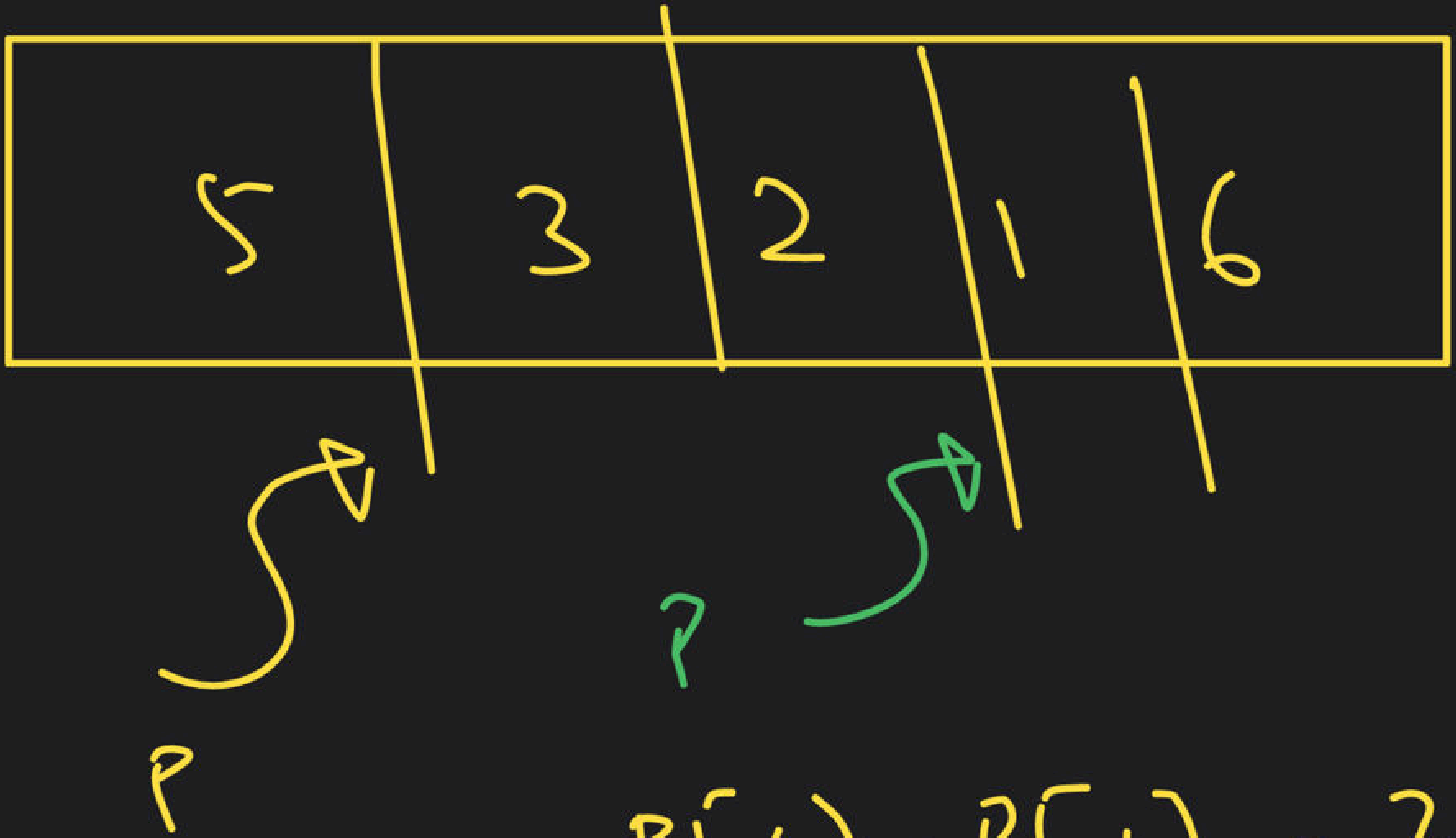


$p[0] - p[1]$

$\Downarrow$

$*(p+1)$

unacademy



$P[0) - P[1] \Rightarrow 3 - 2$

$P[0) - P[1) \Rightarrow \boxed{-5}$

unacademy

Q3.

unacademy

unacademy

THANK YOU!

Here’s to a cracking journey ahead!