Arrays & Pointers Part - VI

Comprehensive Course on C- Programming

Pankaj Sharma • Lesson 24 • May 31, 2024

unacademy

CS & IT Engineering

C Programming

Arrays & Pointers-VI

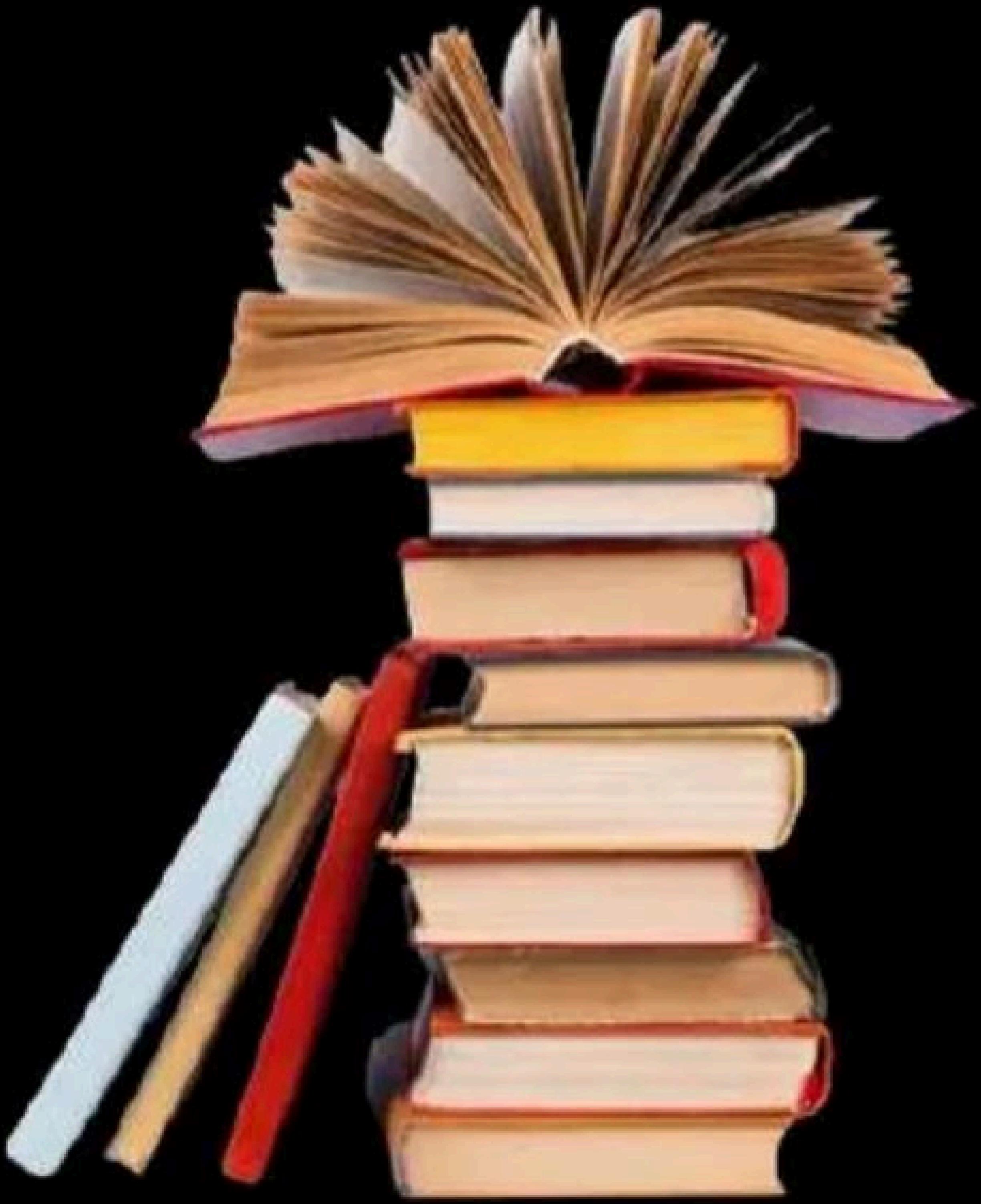
Lecture Number- 24

By- Pankaj Sir

unacademy

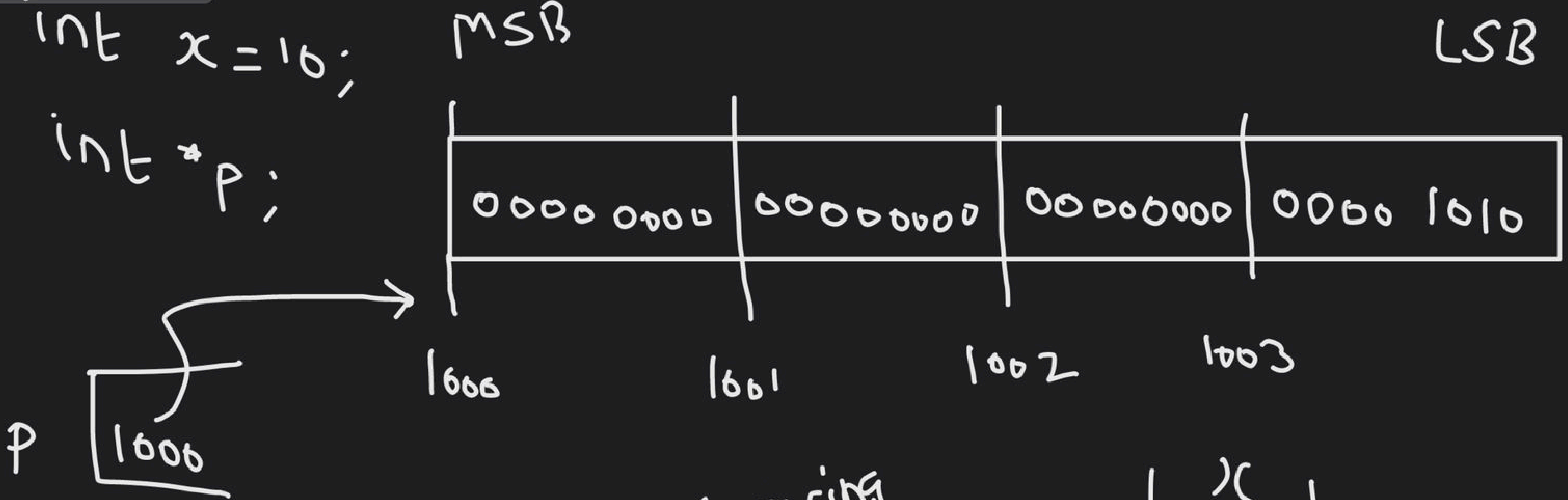
Topics

to be covered

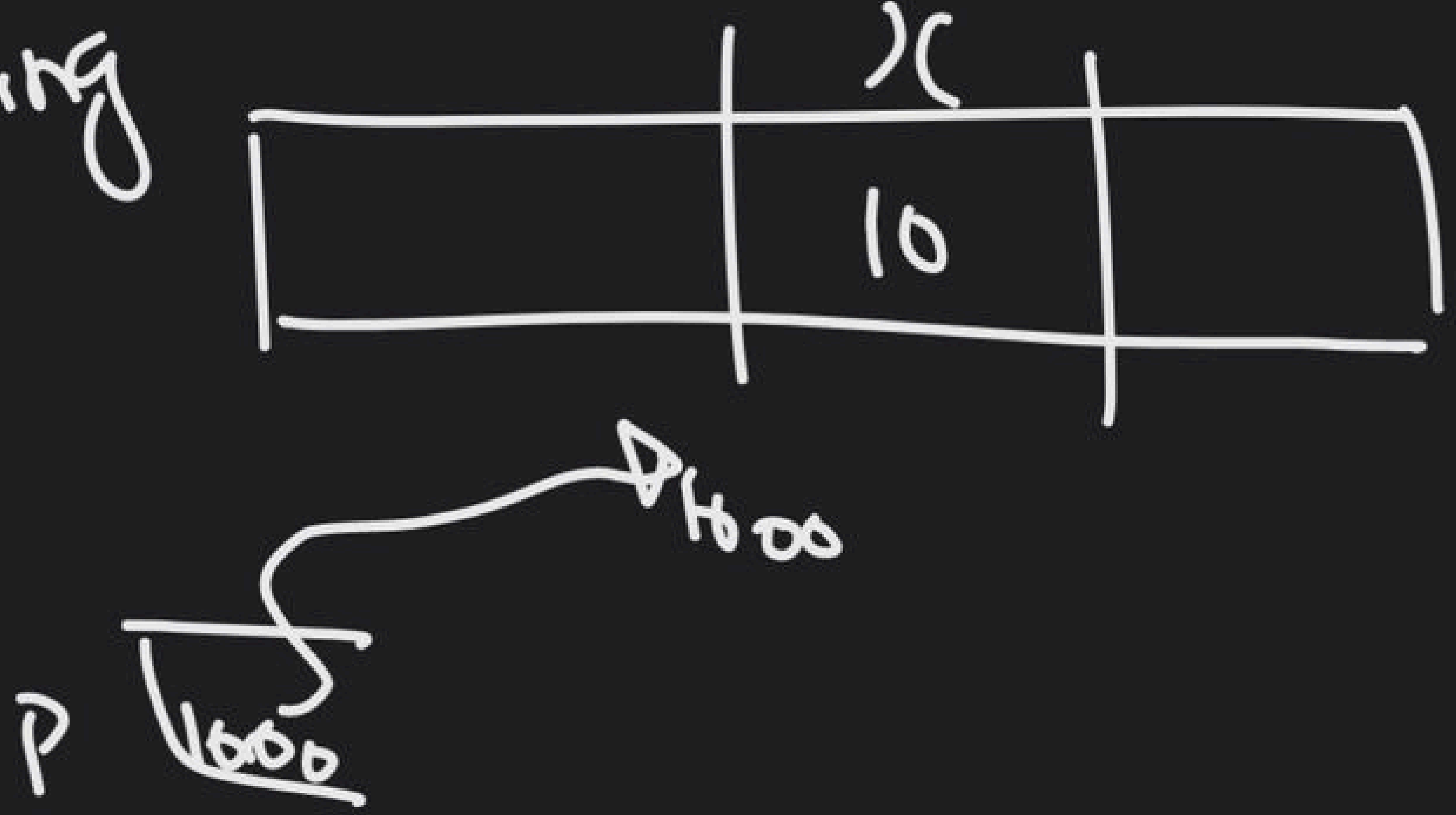


1. Arrays & Pointers Part-VI

```
int x = 10;
int *p;
```



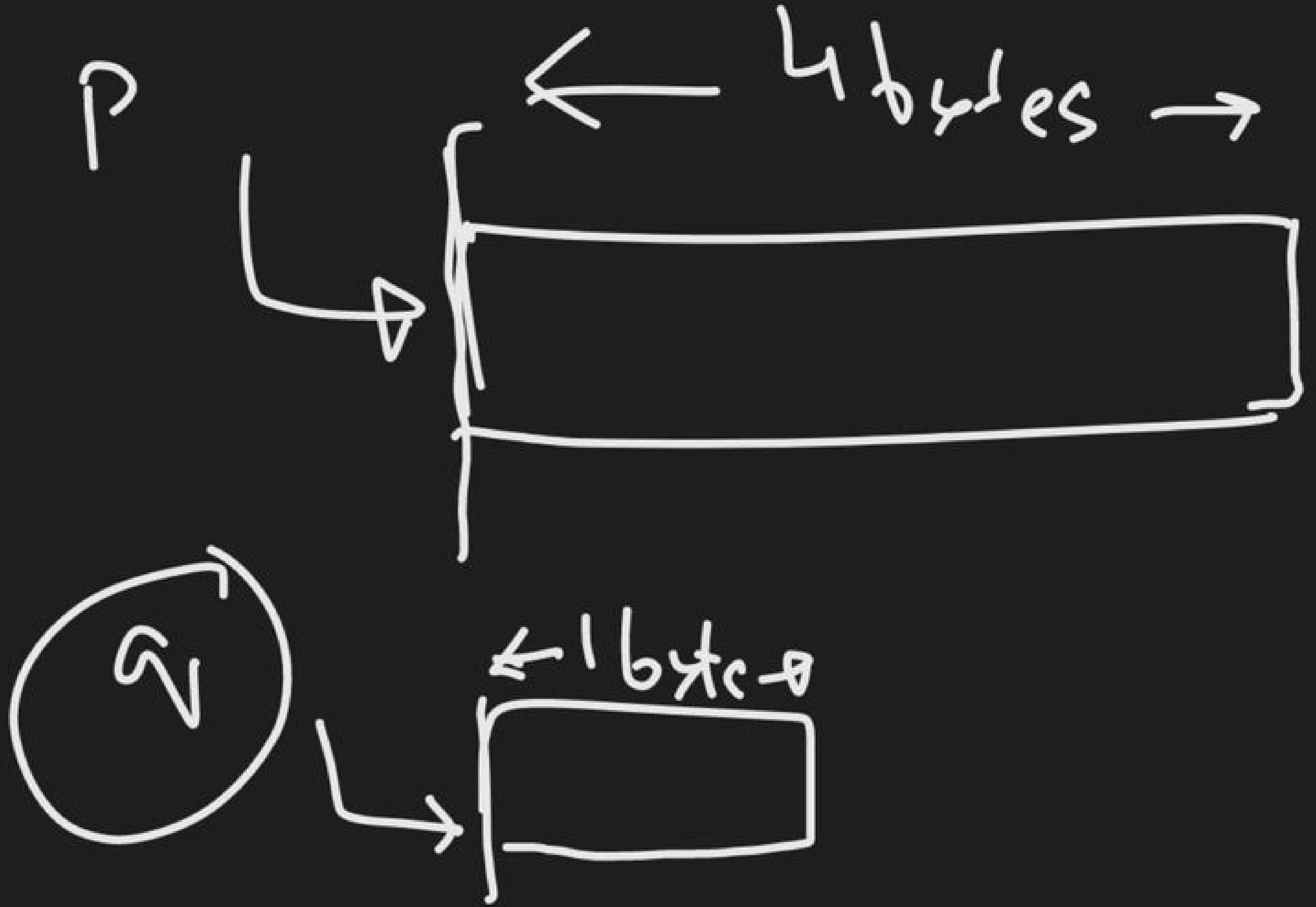
```
pf("%d", *p);
```



int *p → 4 → declaration

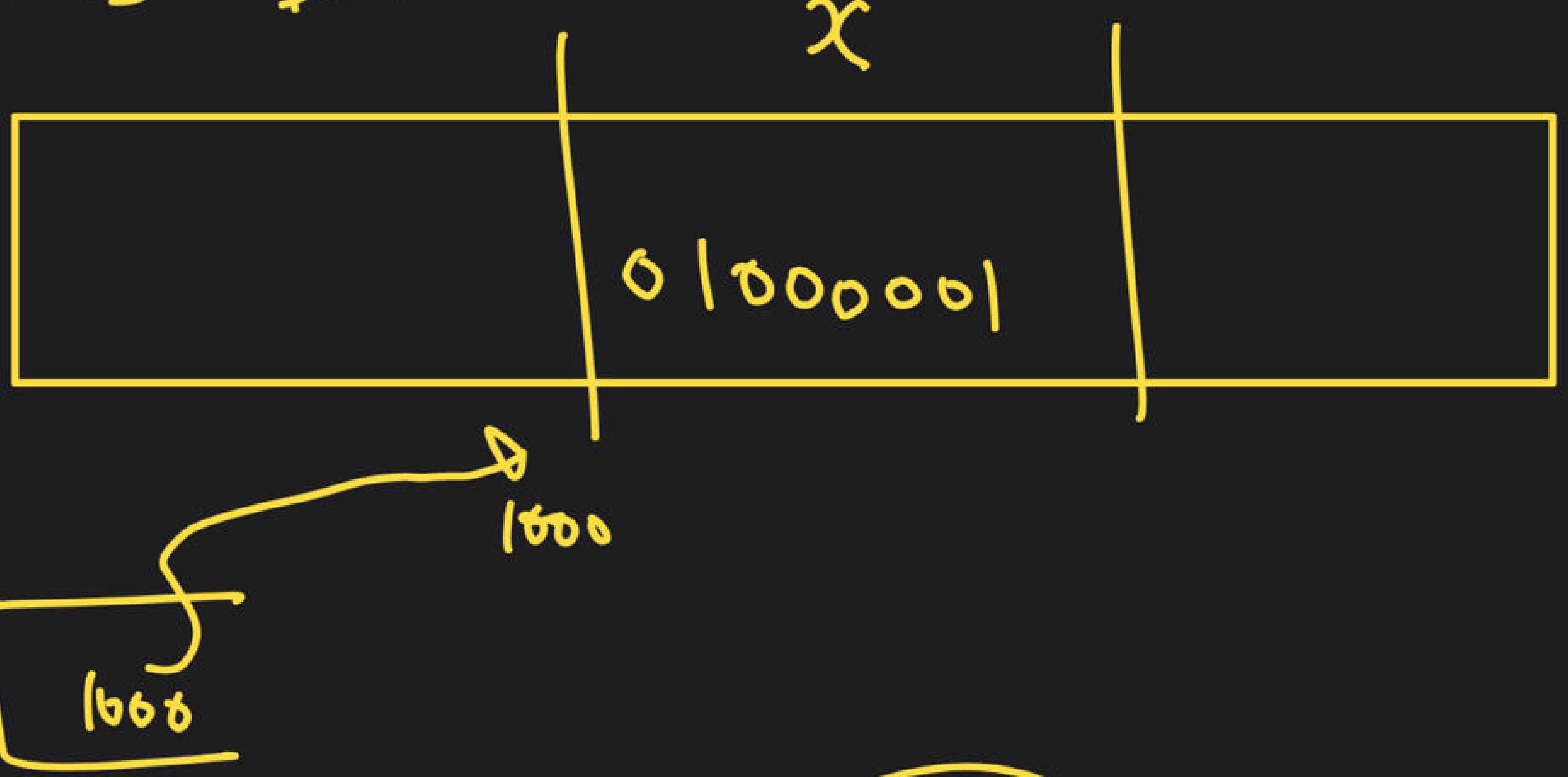
char *q; → 1 byte

pf("%d", *p);



Ex2.

```
char x = 65;   → 01000001
char * p;     // p is a pointer to char

p = &x

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



$\boxed{1\text{byte}}$

$\boxed{65}$

int *p ; p → address of integer variable

Address + val = Add.

value + Add → Add

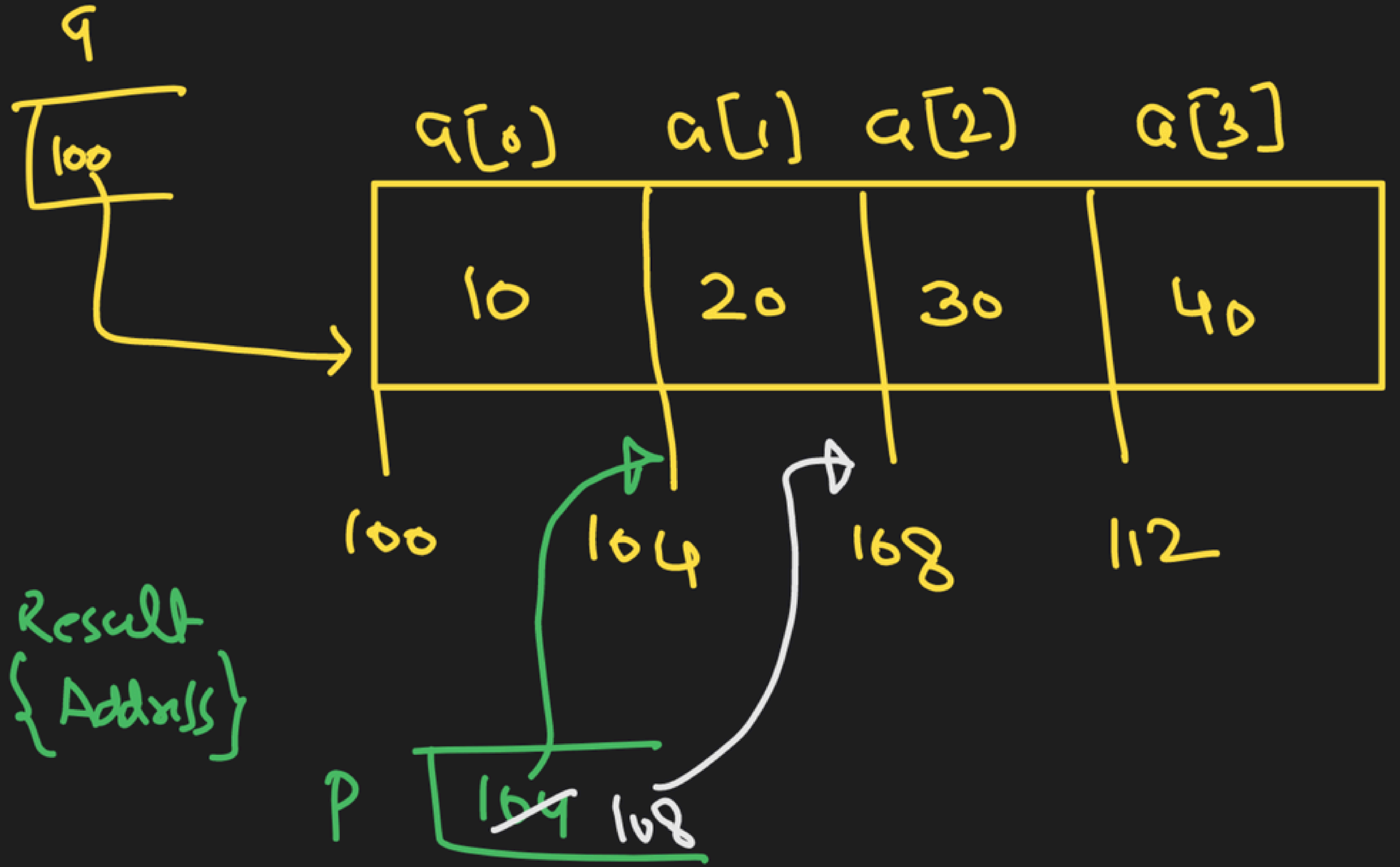
Declaration

```
int a[4] = {10,20,30,40};

int *p;

P= &a[1];

P = P+1 ;
```

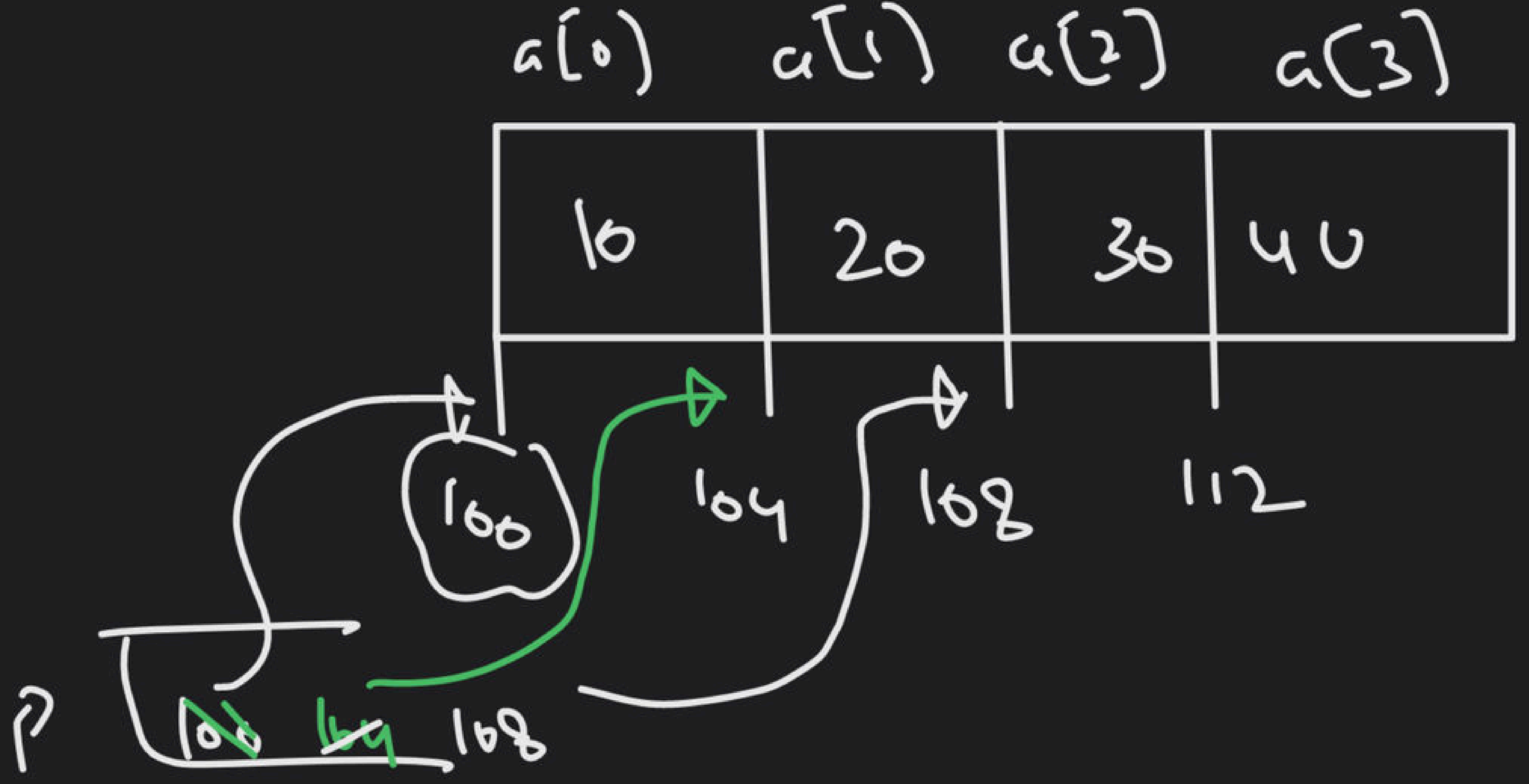
Address + value {Address} Result

$P = 104 + 1 \times 4$

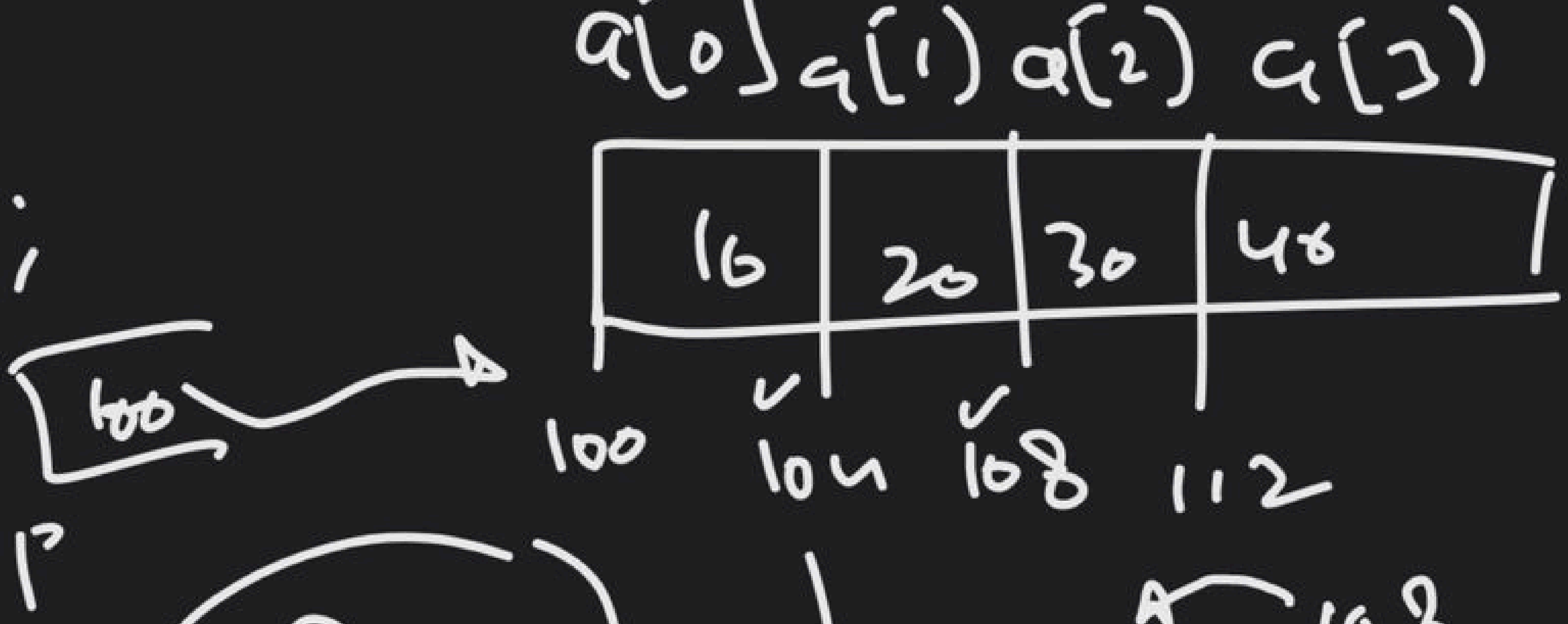
$= 104+4 = 108$



```
int a[4] = {10,20,30,40};
int *p;
p = &a[0];
p++; ✓
++p;
```



```
int a[4] = {10,20,30,40};
```



```
int *p = a;
```

OR

```
int *p;
p = a;
```

```
p+2;
```

108

p $\boxed{100}$

No updation in p

```
p = p+2;
```

108

p $\boxed{\cancel{100}\ 108}$

```
int x = 12;
```

$$\boxed{\begin{array}{l}\texttt{int x;}\\ \texttt{x = 12;}\end{array}}$$

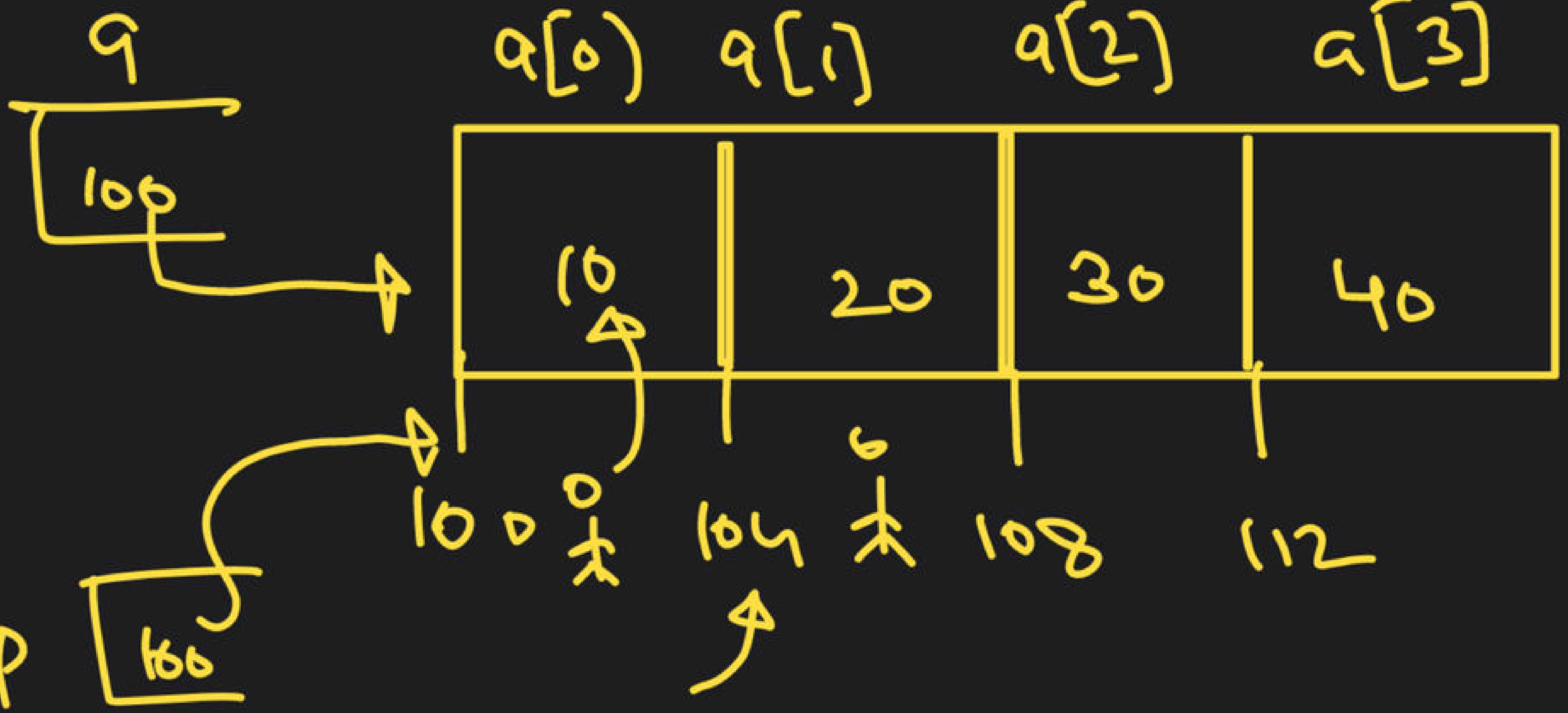
$$\downarrow$$

```
int x = 12;
```

$$\boxed{\begin{array}{l}\texttt{int *p;}\\ \texttt{p = a;}\end{array}}$$

$$\downarrow$$

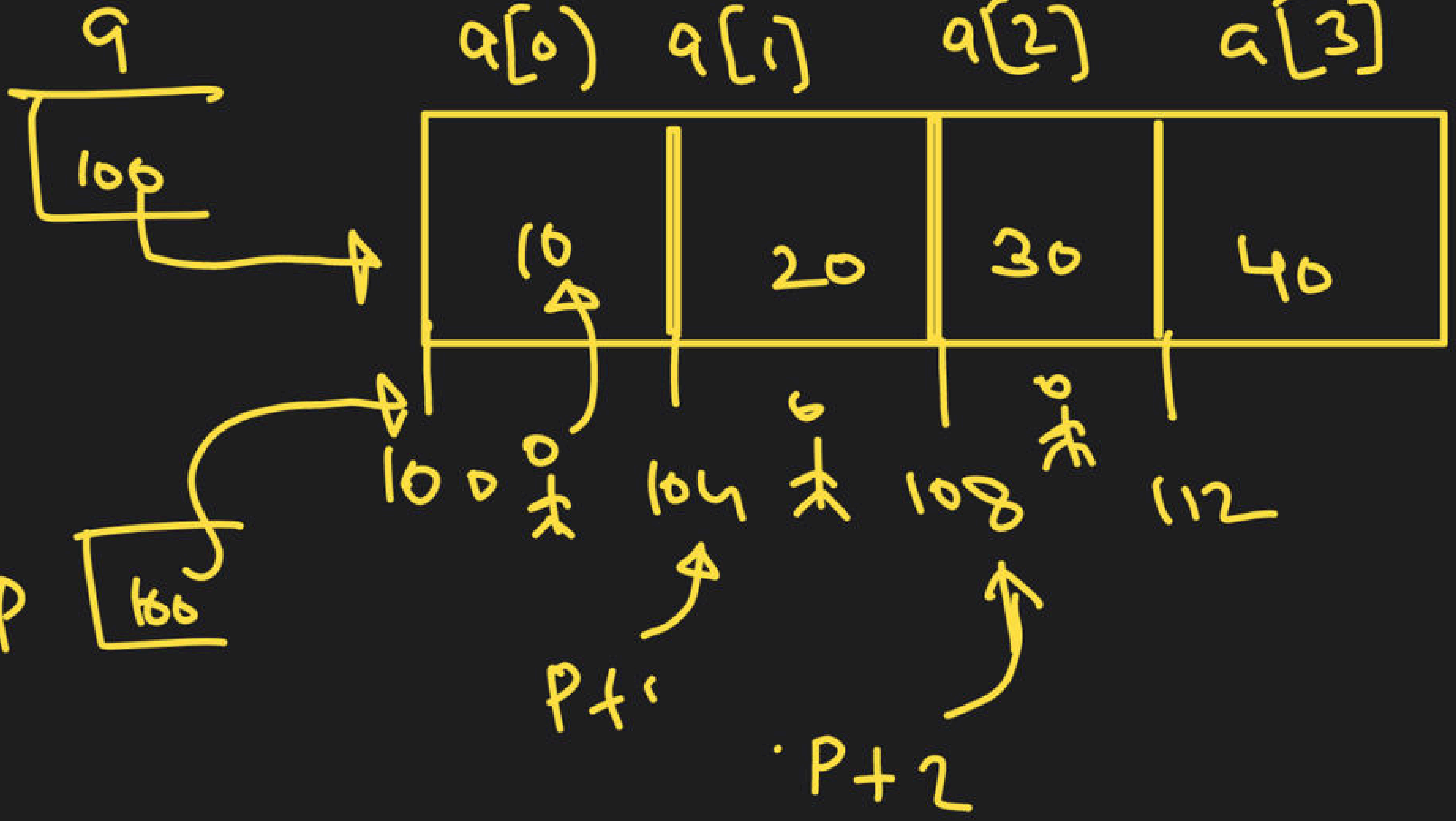
```
int *p = a;
int a[4] = {10,20,30,40};
int *p;
p = a;

pf("%d", *p); 10
pf("%d", *(p+1)); 20
```



p+1 ⇒ Memory loc. 104

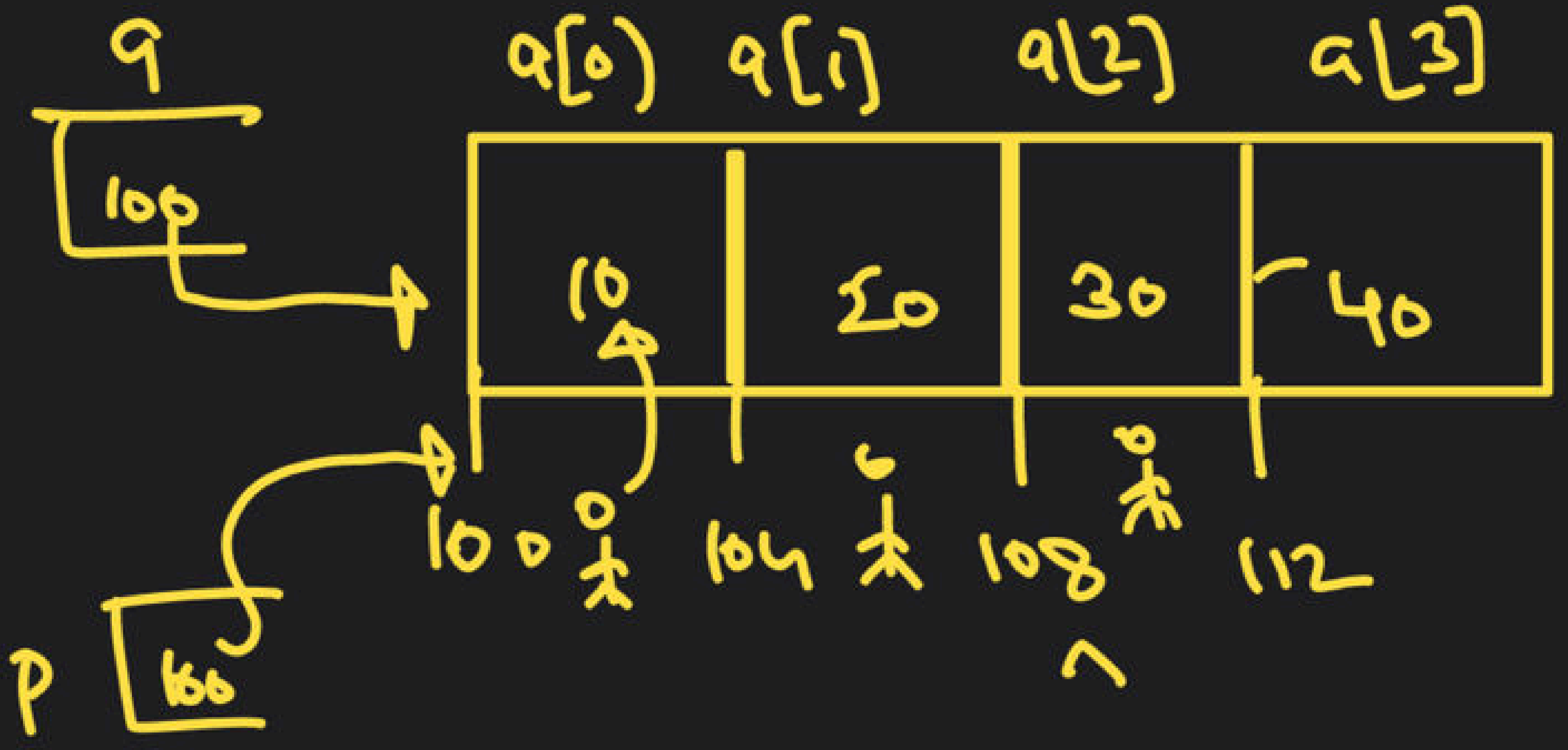
*(p+1) = value at (Mem. loc. 104)

```
int a[4] = {10,20,30,40};
int *p;
p = a;

pf("%d", *p); 10
pf("%d", *(p+1)); 20
pf("%d", *(p+2)); 30
pf("%d", *(p+3)); 40
```



$*(p+2) \Rightarrow 30$

```
int a[4] = {10,20,30,40};
int *p;
p = a;
pf("%d", *(p+0));
pf("%d", *(p+1)); 20
pf("%d", *(p+2)); 30
pf("%d", *(p+3)); 40

pf("%d", p[0]);
pf("%d", p[1]);
pf("%d", p[2]);
pf("%d", p[
```



$p \rightarrow p+0$ ✓

$*(p+i) = p[i]$

```
pf("%d", p[0]);
pf("%d", p[1]);
pf("%d", p[2]);
pf("%d", p[3]);
```

→ What is p → array

✓

```
int *p;
p = a
```

```
int *p = a;
```

→ declar

```
✓ int x;
✓ x = 12;
```

```
→ int x = 12;
```

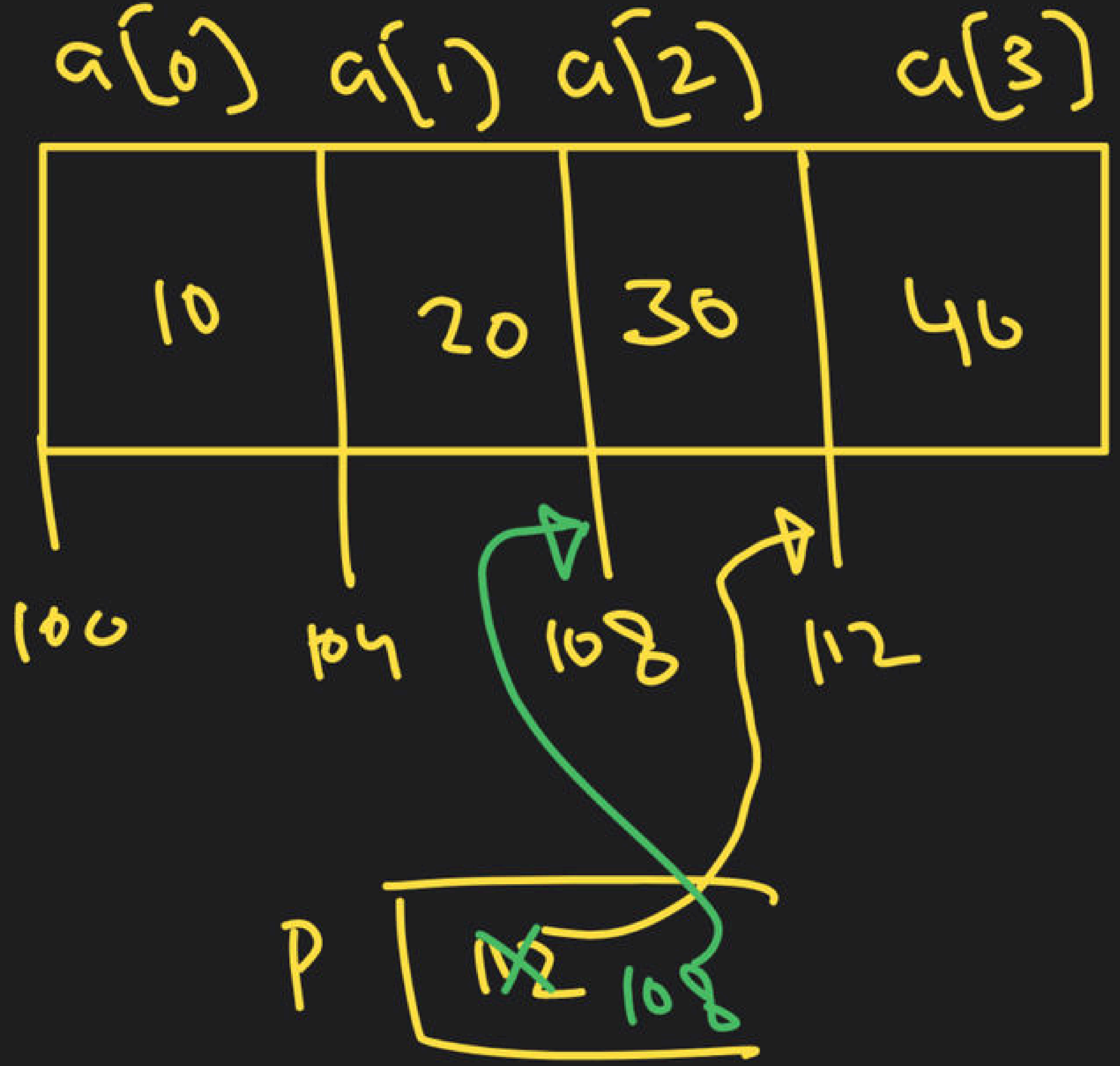
```
int a[4] = {10,20,30,40};
```

$$\left.\begin{array}{l}\texttt{a++;}\\ \texttt{++a;}\\ \texttt{--a;}\\ \texttt{a--;}\end{array}\right\}\text{ Invalid}$$

```
int a[4] = {10,20,30,40};
            ↑  ↑
int *p = a;
p++;   ✓
++p;   ✓
```

Ex

```
int a[4] = {10,20,30,40};
    int *p = &a[3];
    --p;
```



1. Pointer + Pointer
2. ++Ptr, --Ptr, Ptr++, Ptr--
3. Ptr + 3 ⇒ valid (Moving 3 locations in forward direction)
4. Ptr - 3 → valid (Moving 3 loc. in backward direction)

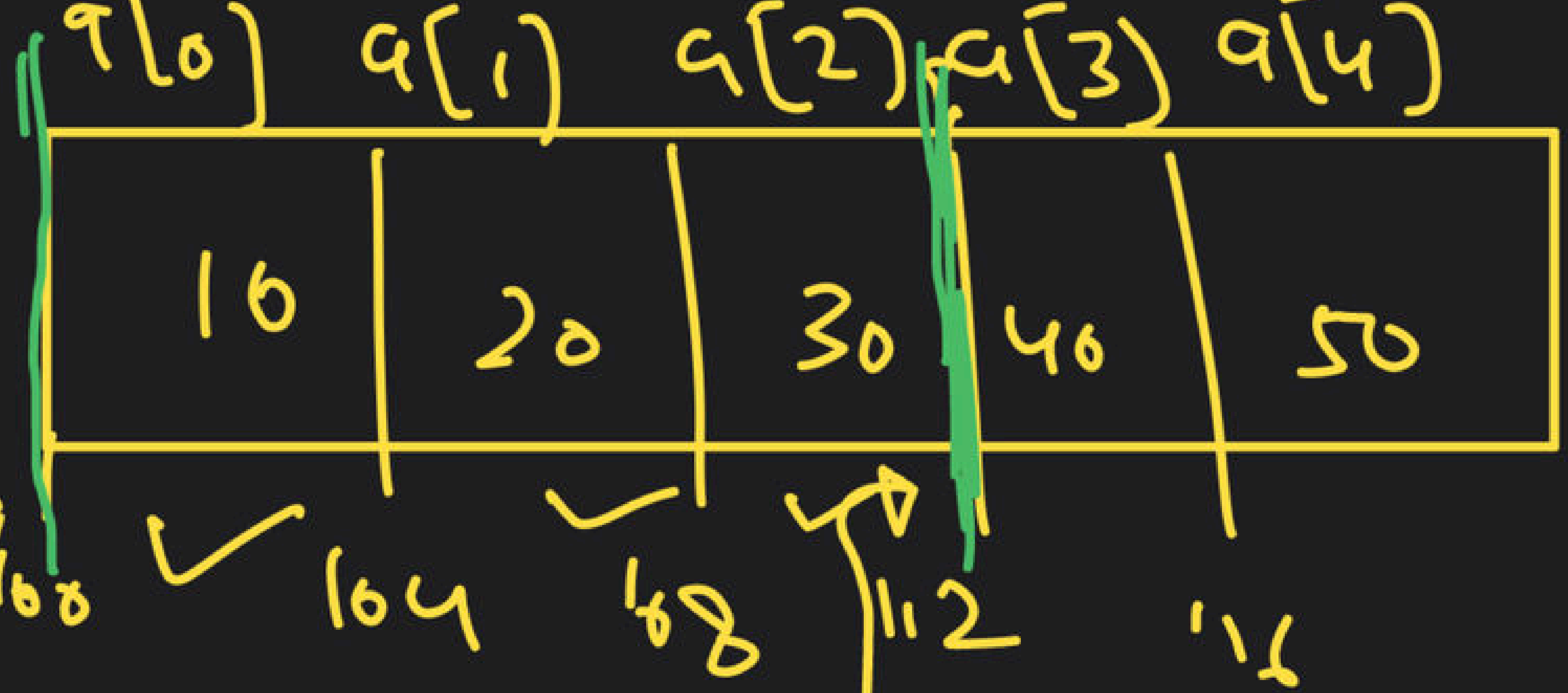
unacademy

Ptr1 - Ptr2 $\Rightarrow$ Only in 1 case it is valid

int *p, *q;

p = &a[3];

q = &a[0]



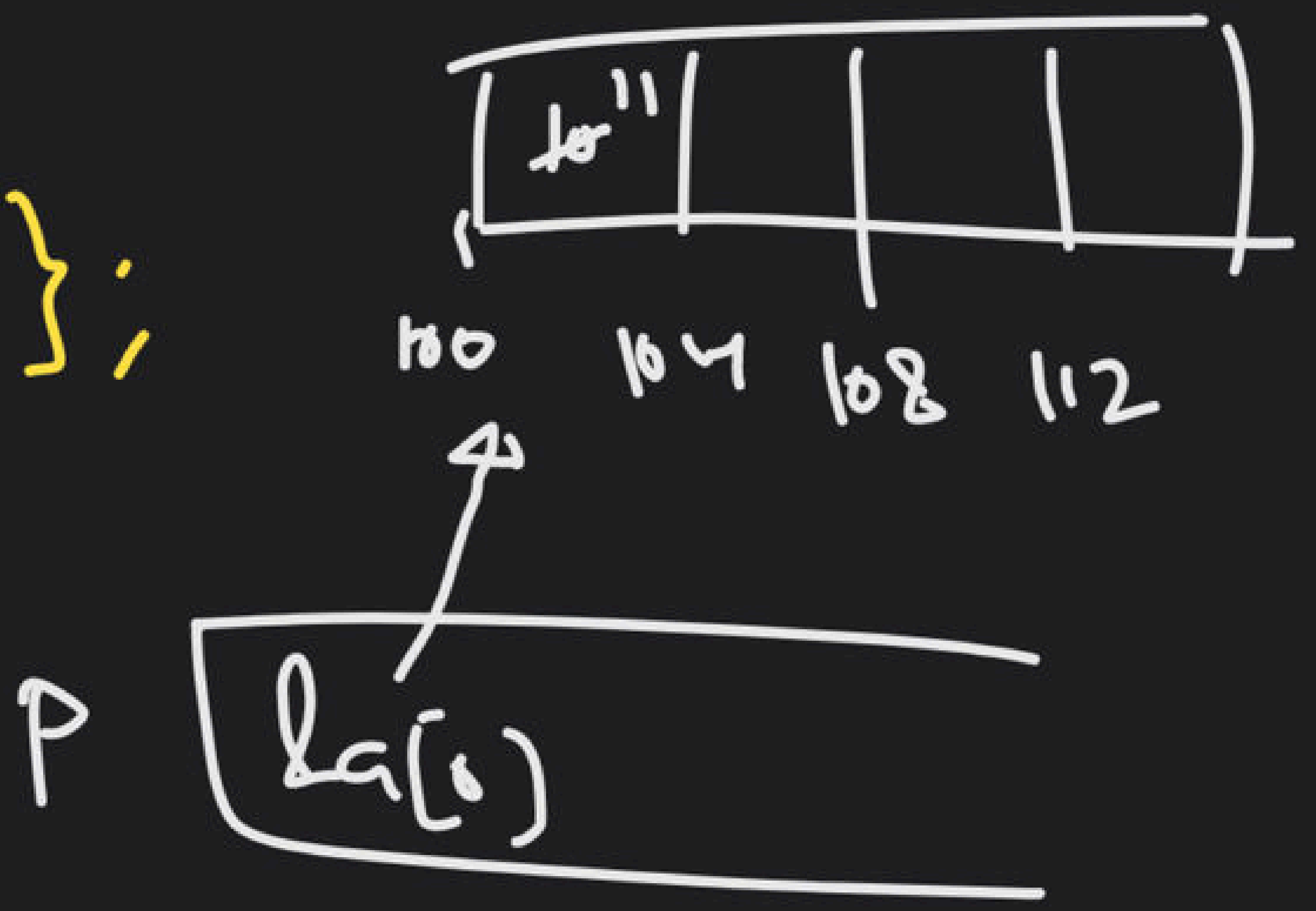
P-q ✓ $= \frac{(\text{Actual diff})}{\text{int size}} = \frac{112-100}{4} = \frac{12}{4} = \boxed{3}$

4

++ , *

Q

```
int a[4] = {10,20,30,40};
int *p = &a[0];
✓ ++*p;
✓ *p++;
pf("%d", *p);
*(p++);
```



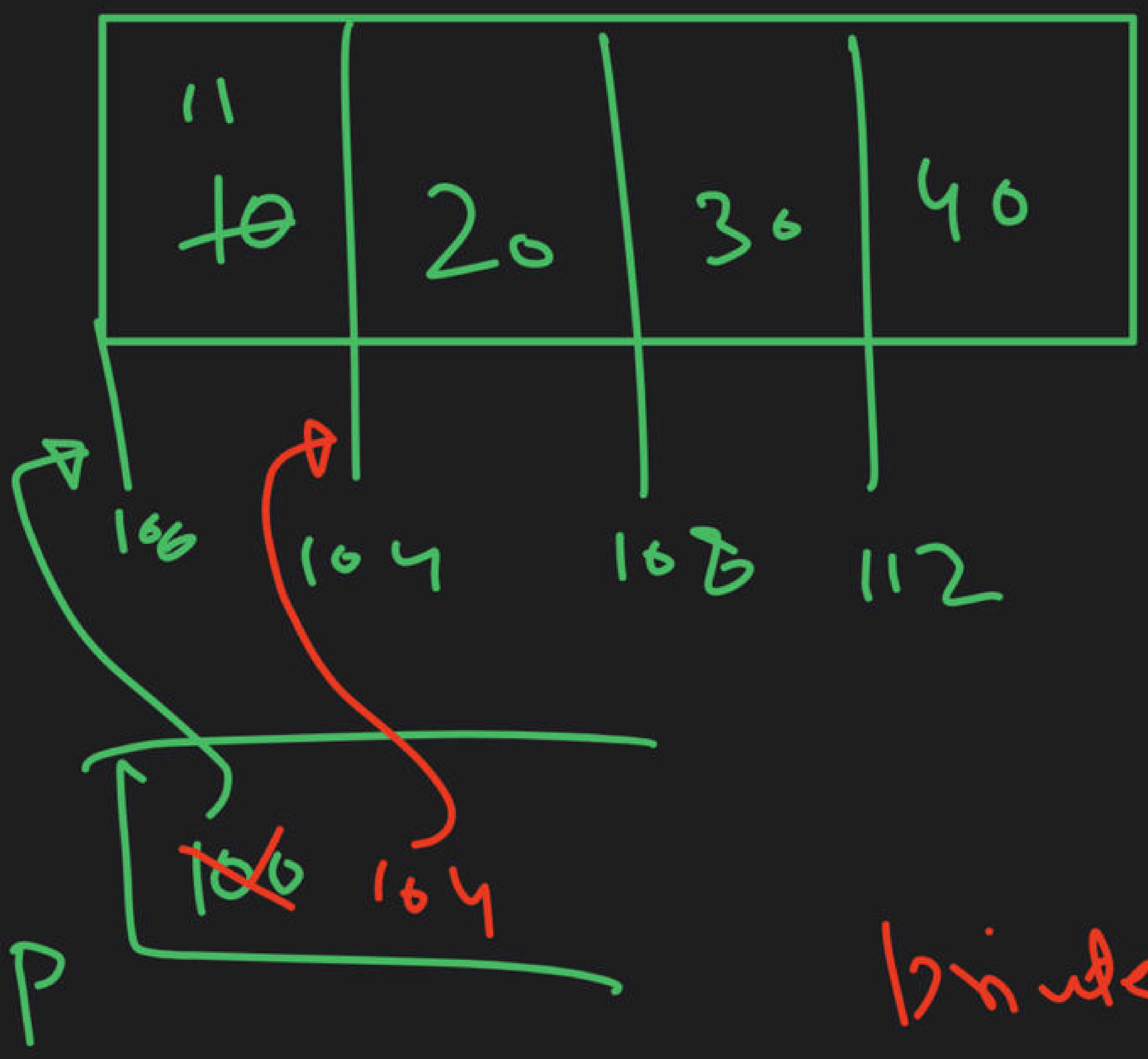
```
++(*p)
(i) *p = *p+1;
(ii) use *p (useless)
```

$*(p{+}{+}) \rightarrow$ Post-inc

i) use $*(p)$

Assign ×, print ×
exp ×

ii) $p = p + 1$



printed

$*p \rightarrow \boxed{20}$

Q

```
int a[4] = {10, 20, 30, 40};
int *p = &a[0];

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

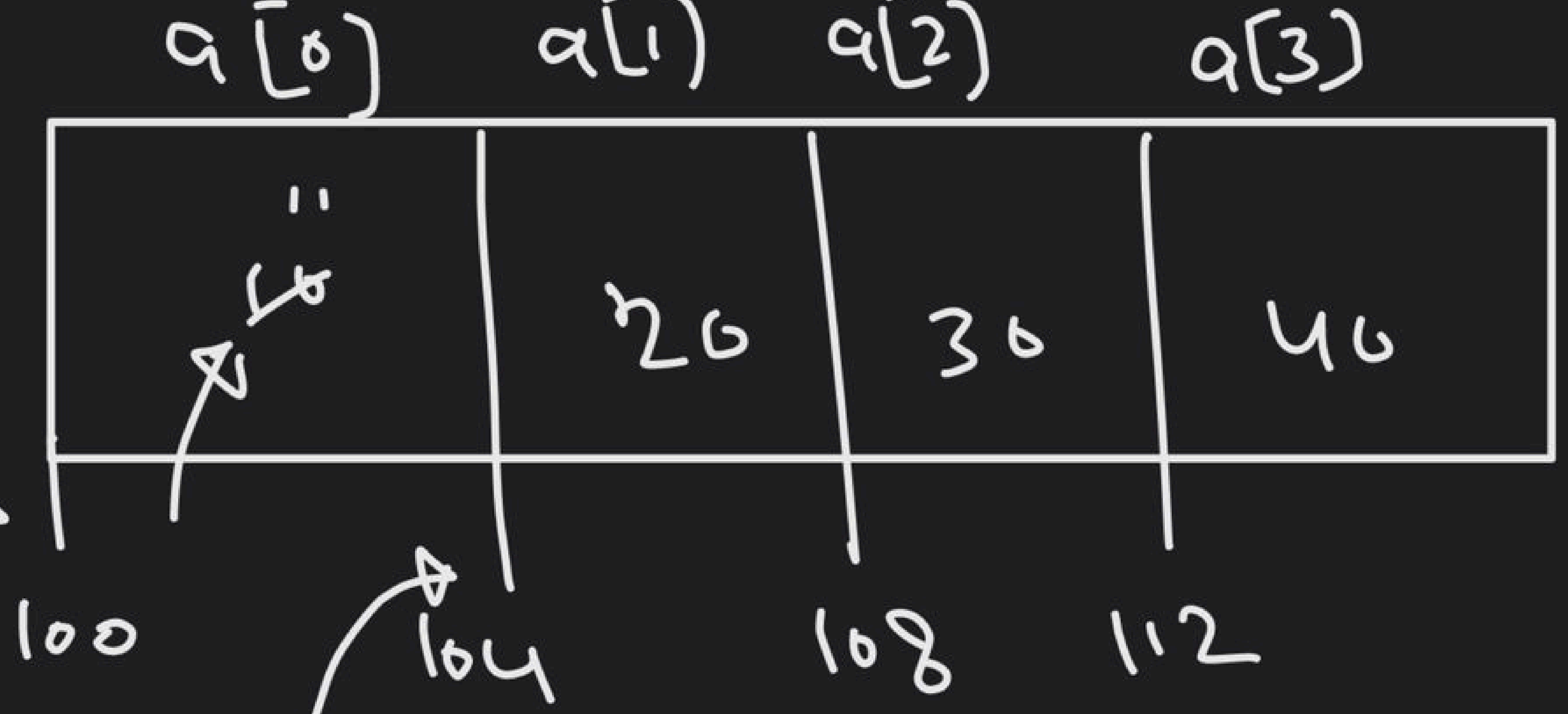
$\rightarrow$ Error

$\underline{++}\,P\,\underline{++}$

$++(*(p++))$

① use $++*p$ → inside bf

② $p = p+1$



$\boxed{11}$

$++(*p)$

a) $*p = *p+1$

b) use $*p$ → pf

unacademy

THANK YOU!

Here’s to a cracking journey ahead!