

Arrays & Pointers Part - VII

Comprehensive Course on C- Programming



CS & IT Engineering

C Programming
Arrays & Pointers -VII



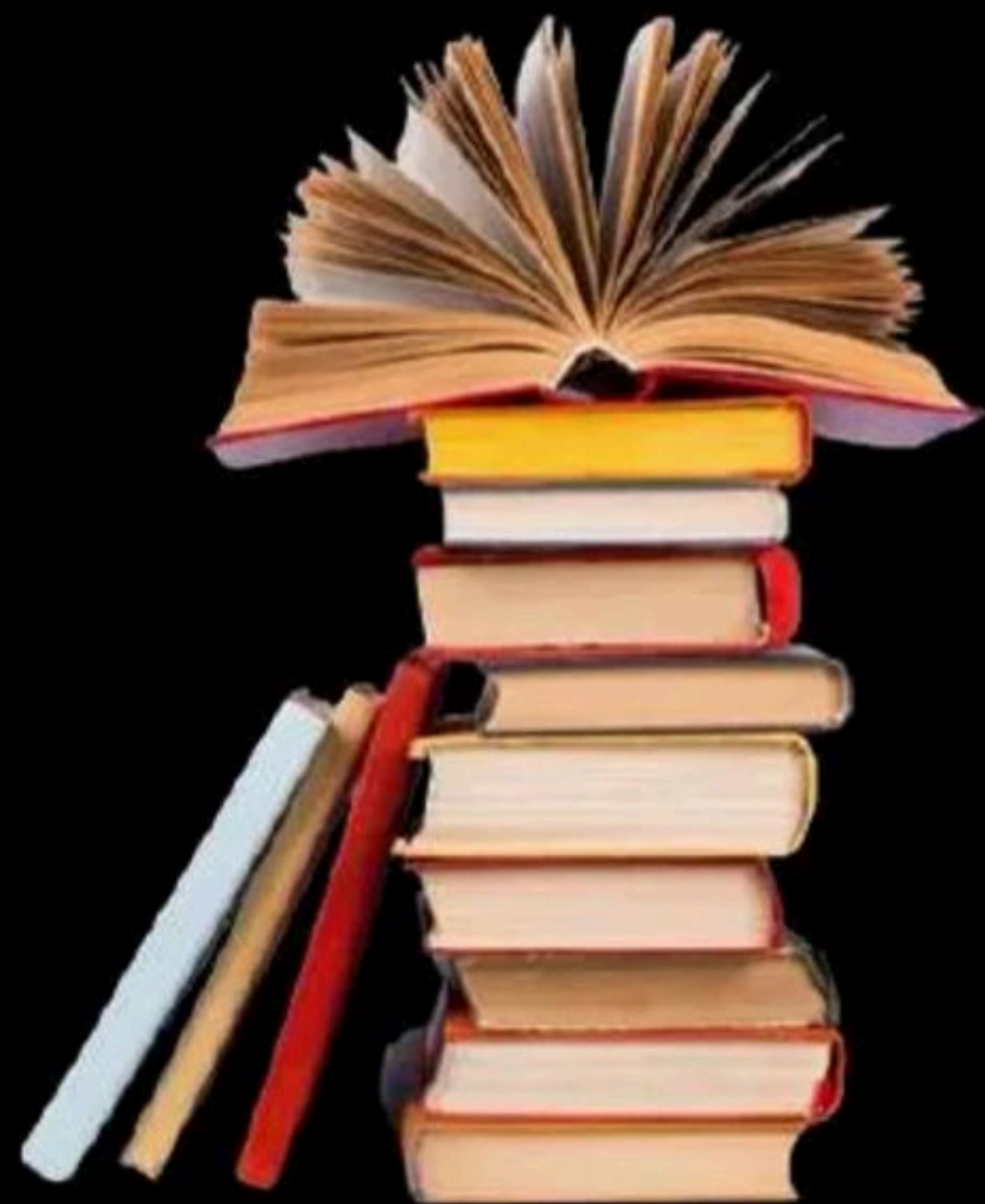
Lecture Number- 25

By- Pankaj Sir



Topics

to be covered



1 Arrays & Pointers Part-VII

Complex Declarations

int * (P[4]);

int (*P)[4];

int (*P)(int, int);

(i)	()	Functions	I	L to R
(ii)	[]	Arrays		
(iii)	*	Pointer	II	R to L
(iv)		Identifier		
(v)		Data type	III	

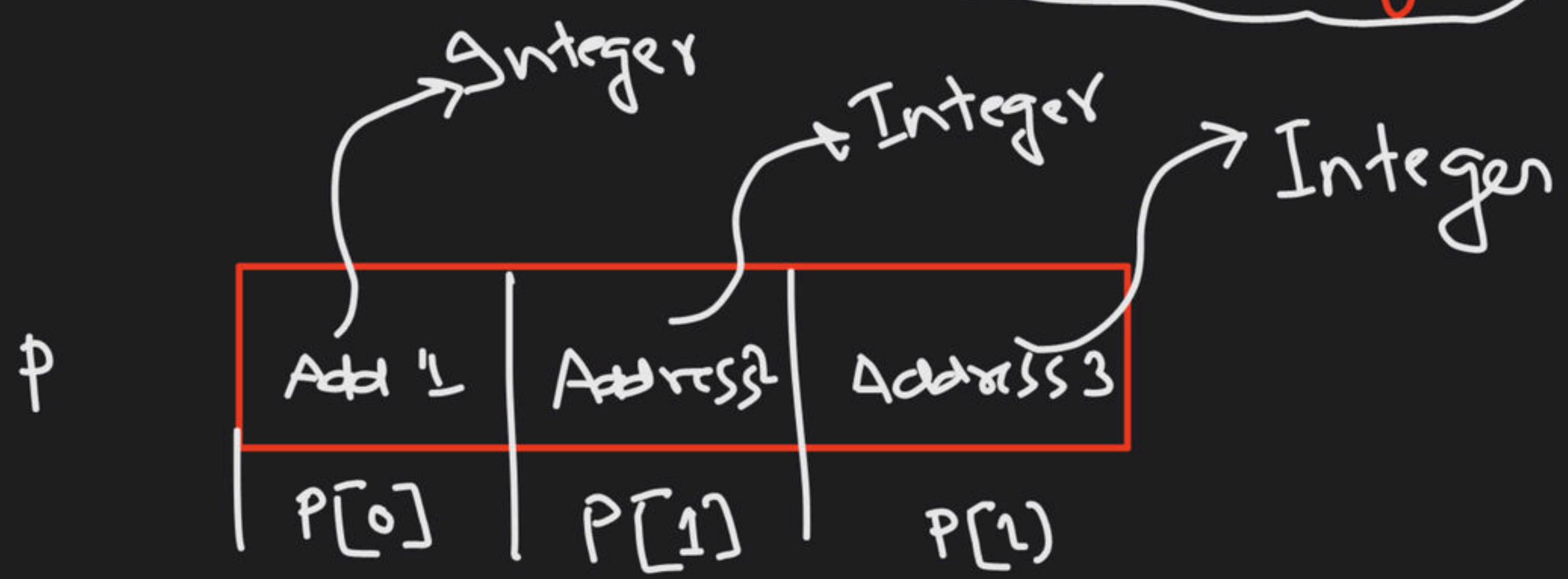
int *p; \Rightarrow

{ int is a star p
star is a int p
p is a _____ }

^③ int ^② * ^① (P[3]);


↓
P is an array of 3 Pointer to integer

P[0] → Pointer to int.
P[1] →
P[2] →



Ex 2.

~~int~~ ~~7~~ ~~P~~ [~~3~~];



⇒ Same

P is array of 3
pointer to integer.

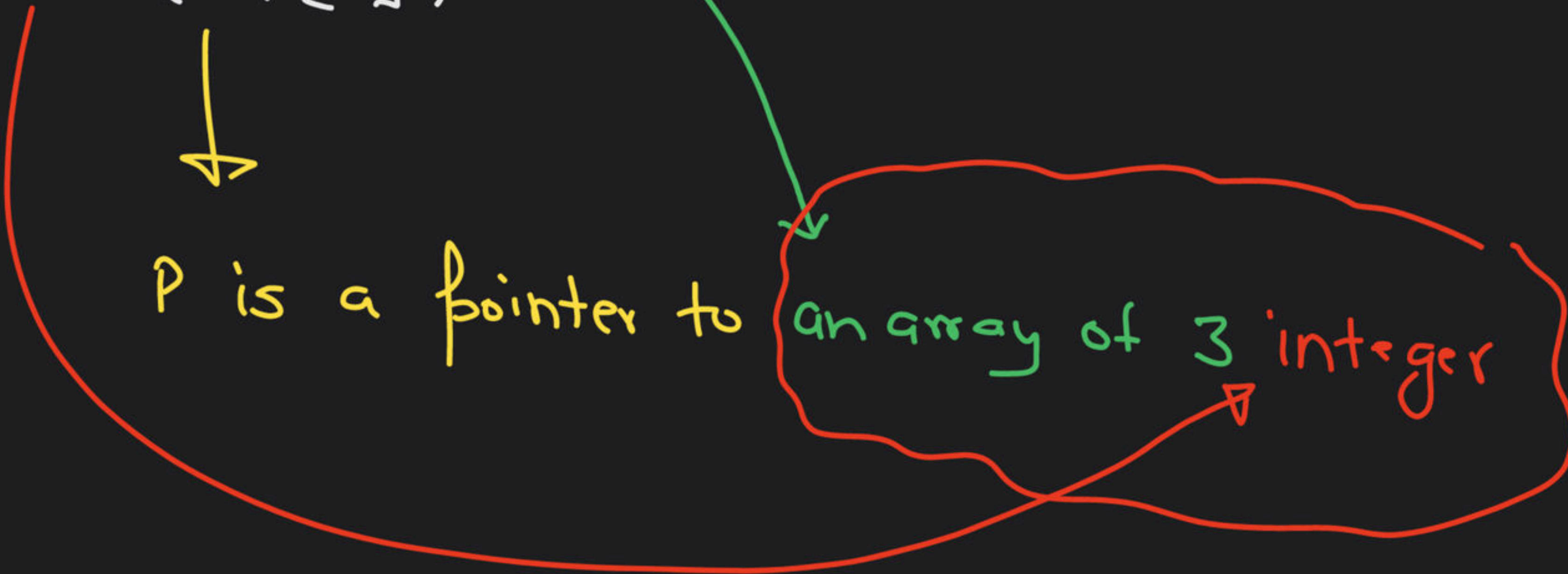
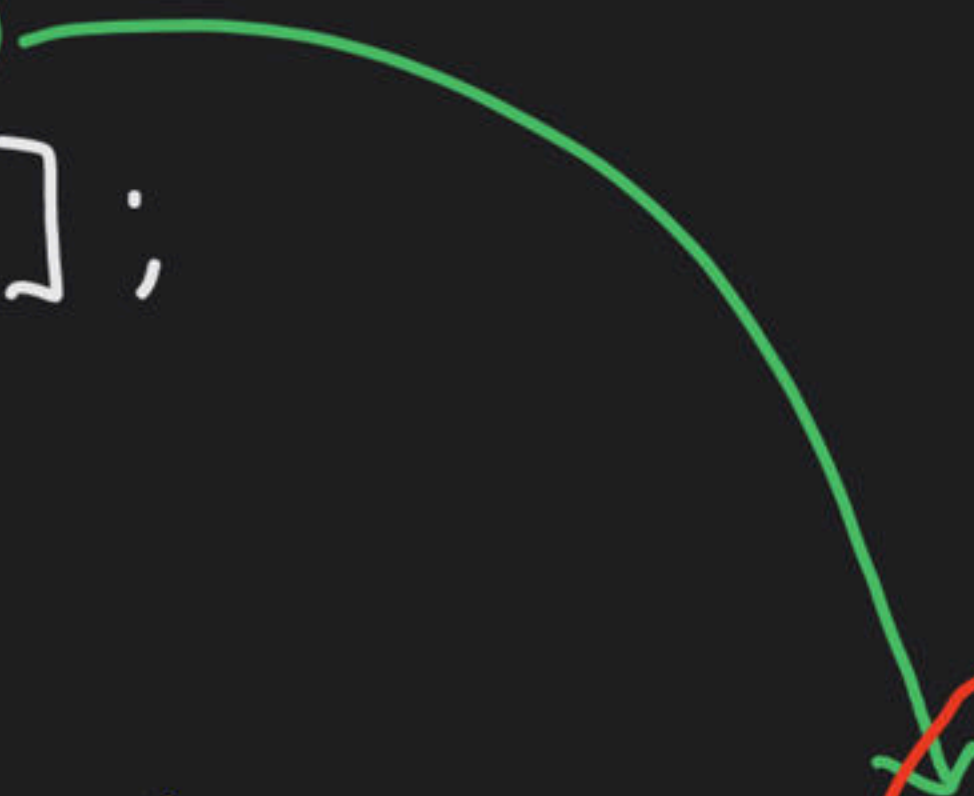
Ex 3.

⁽³⁾
int (*p)[3];



P is a pointer to

an array of 3 integer



P is a pointer to integer.

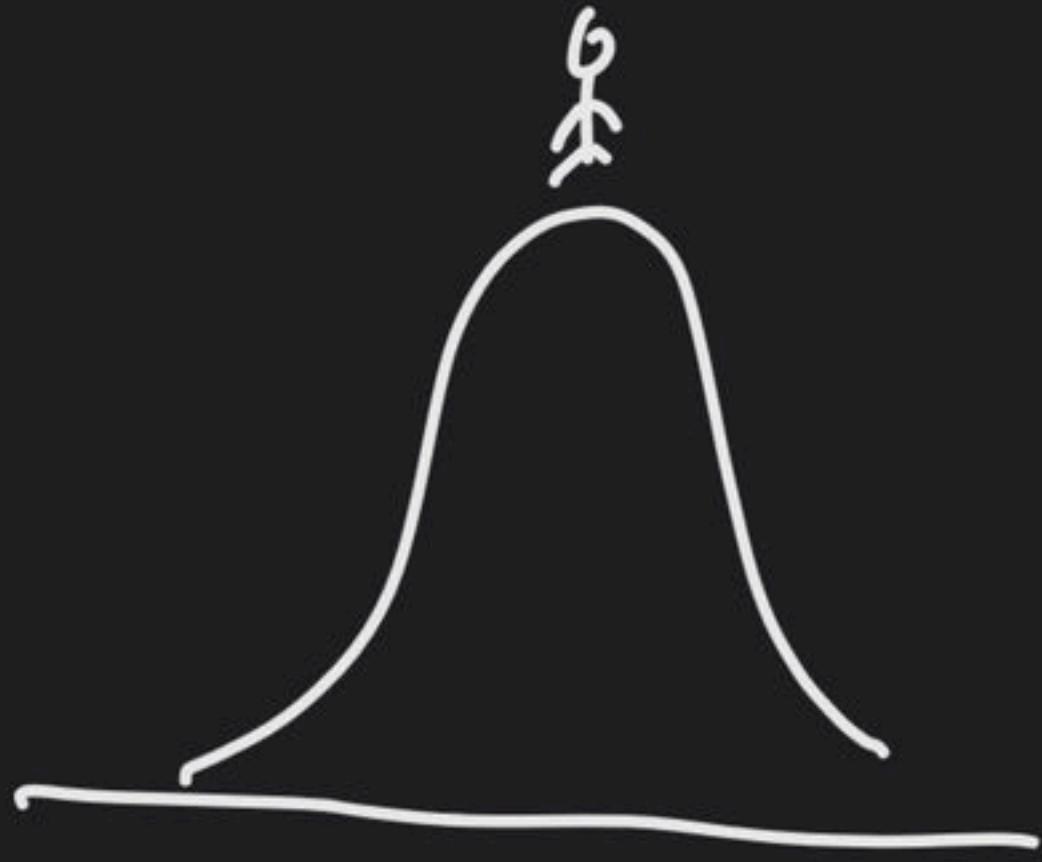
P can hold \rightarrow add of integer

P is a pointer to char

Address of char. var. ✓

P is a pointer to Array of 3 integers

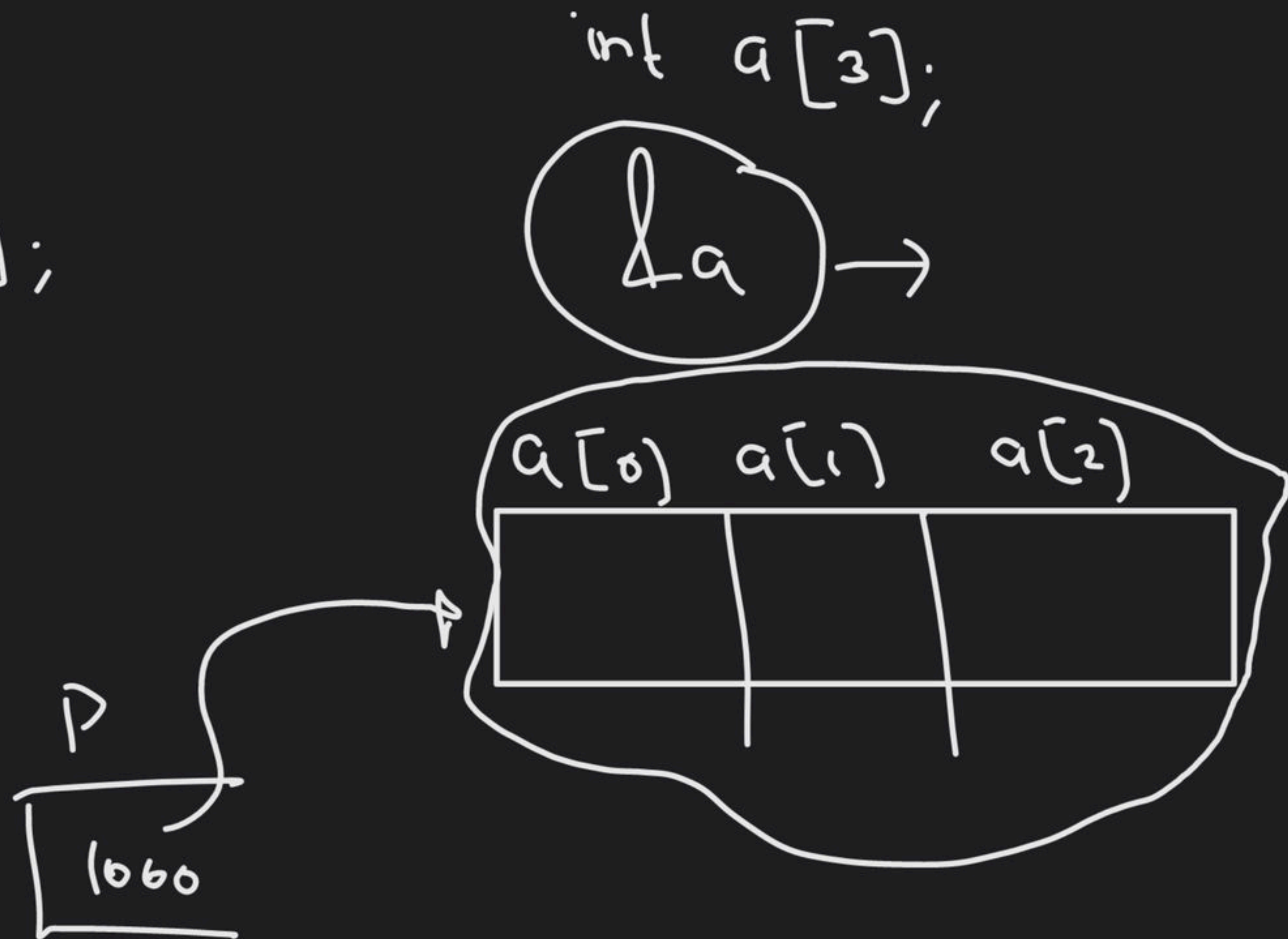
⇒ Address of an array of 3 integers



```
int a[3];
```

```
int (*p)[3];
```

```
p = &a;
```

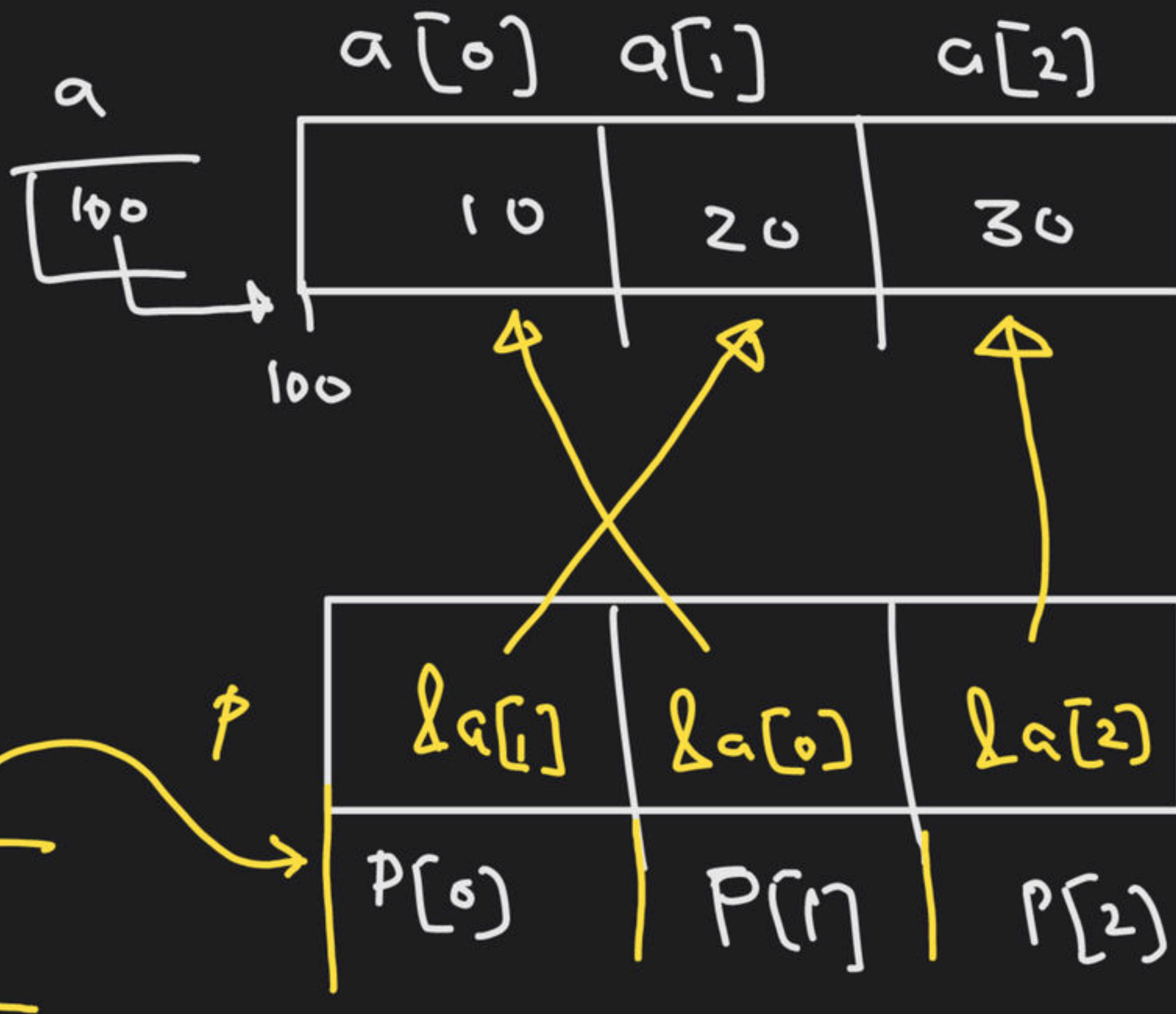


int a[3] = {10, 20, 30};

int *p[3] = {a+1, a, a+2};

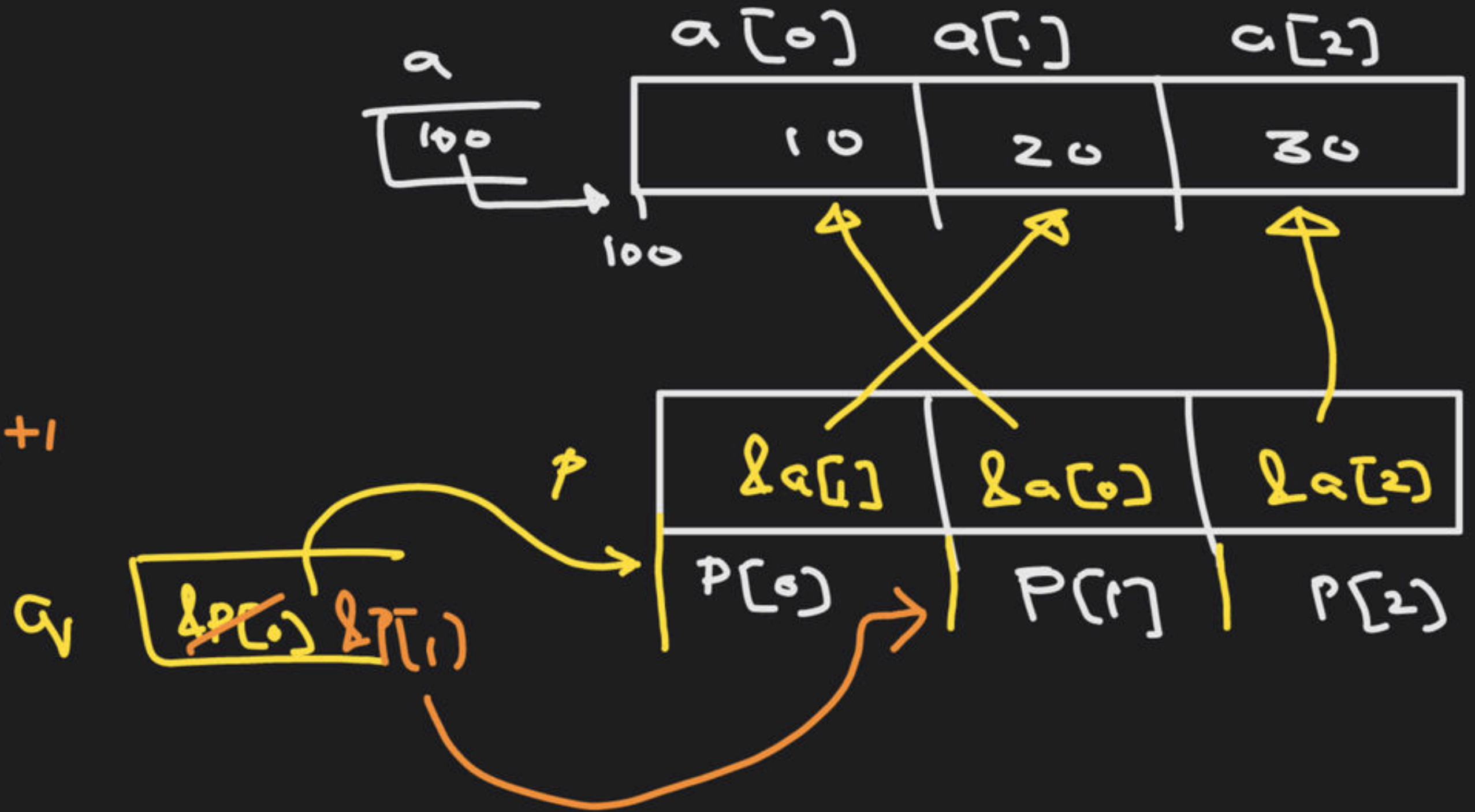
int **q;

① q = p;

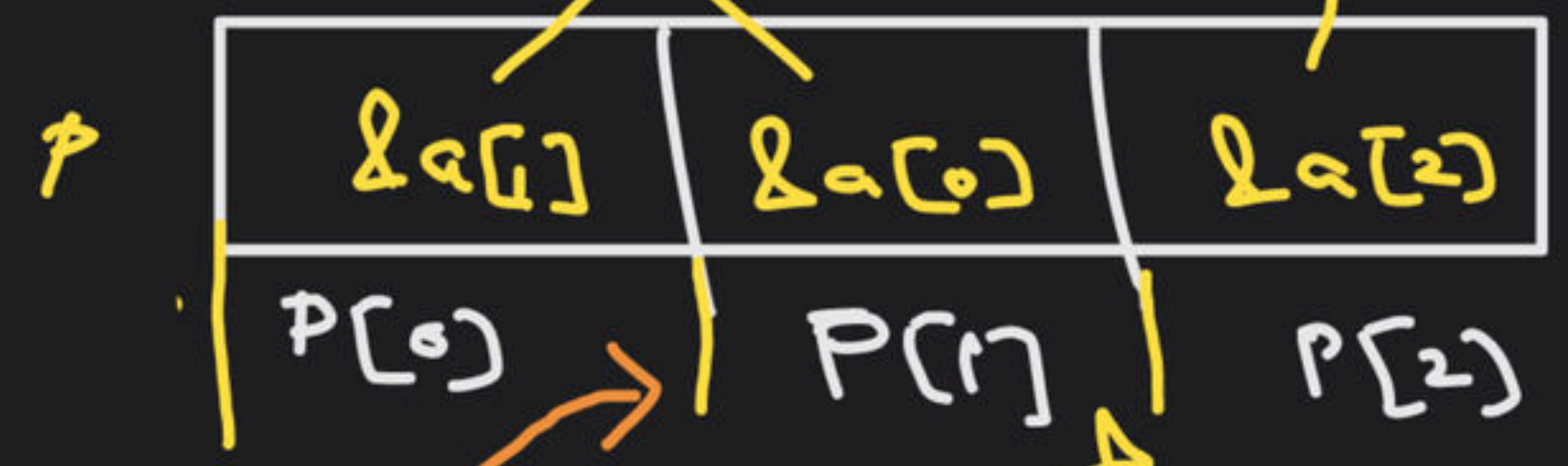
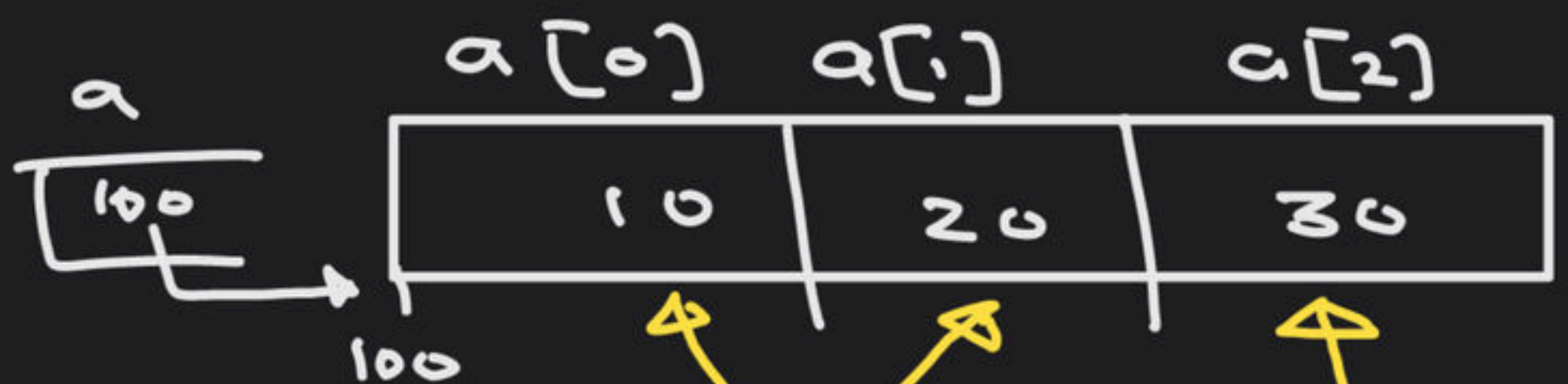


① $q_v = p$;

② $++q_v$; $q_v = q_v + 1$
 $= \&p[0] + 1$
 $= \&p[1]$

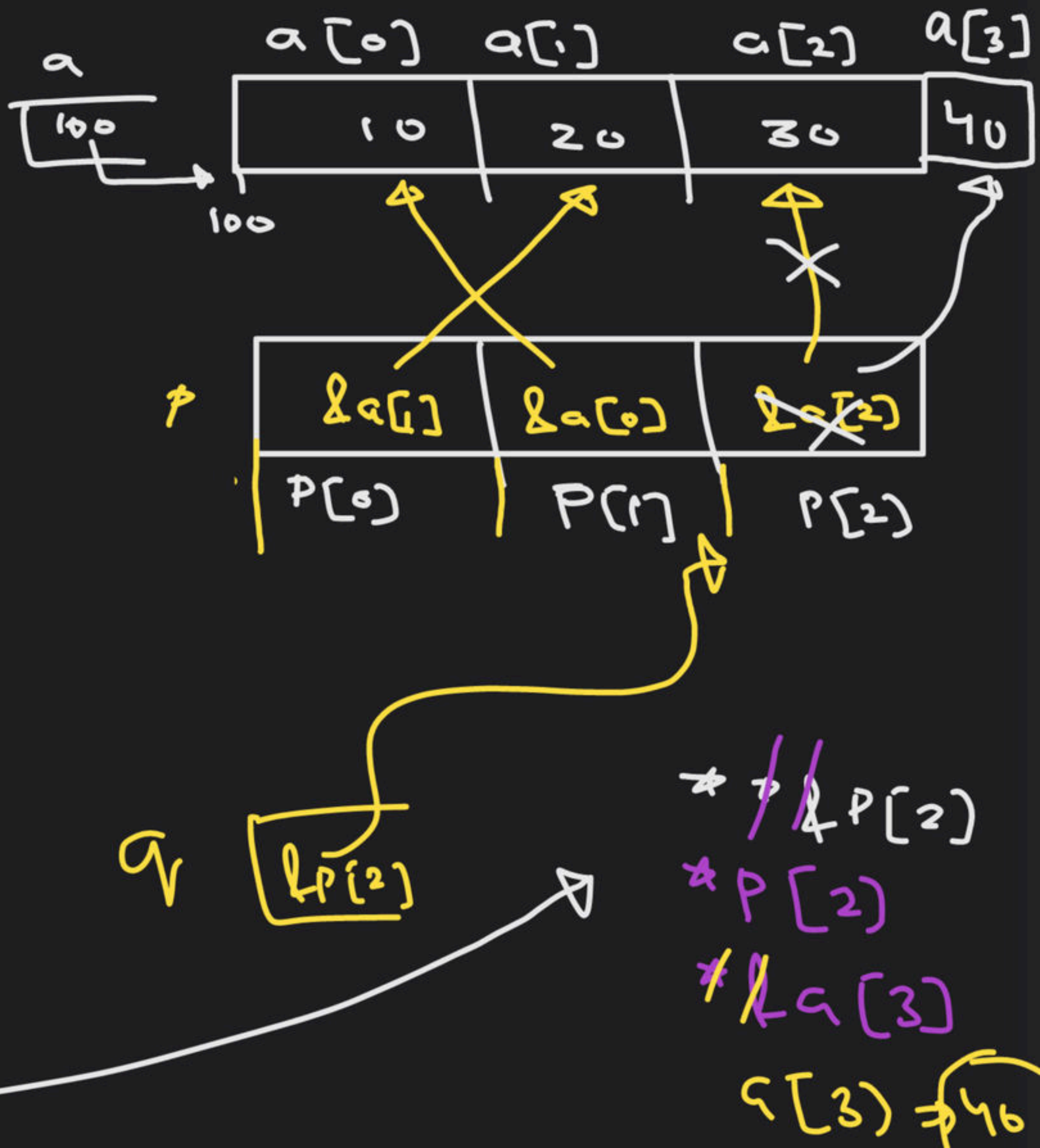


\rightarrow (i) $q = q + 1$
 \rightarrow (ii) use $*q$
No meaning



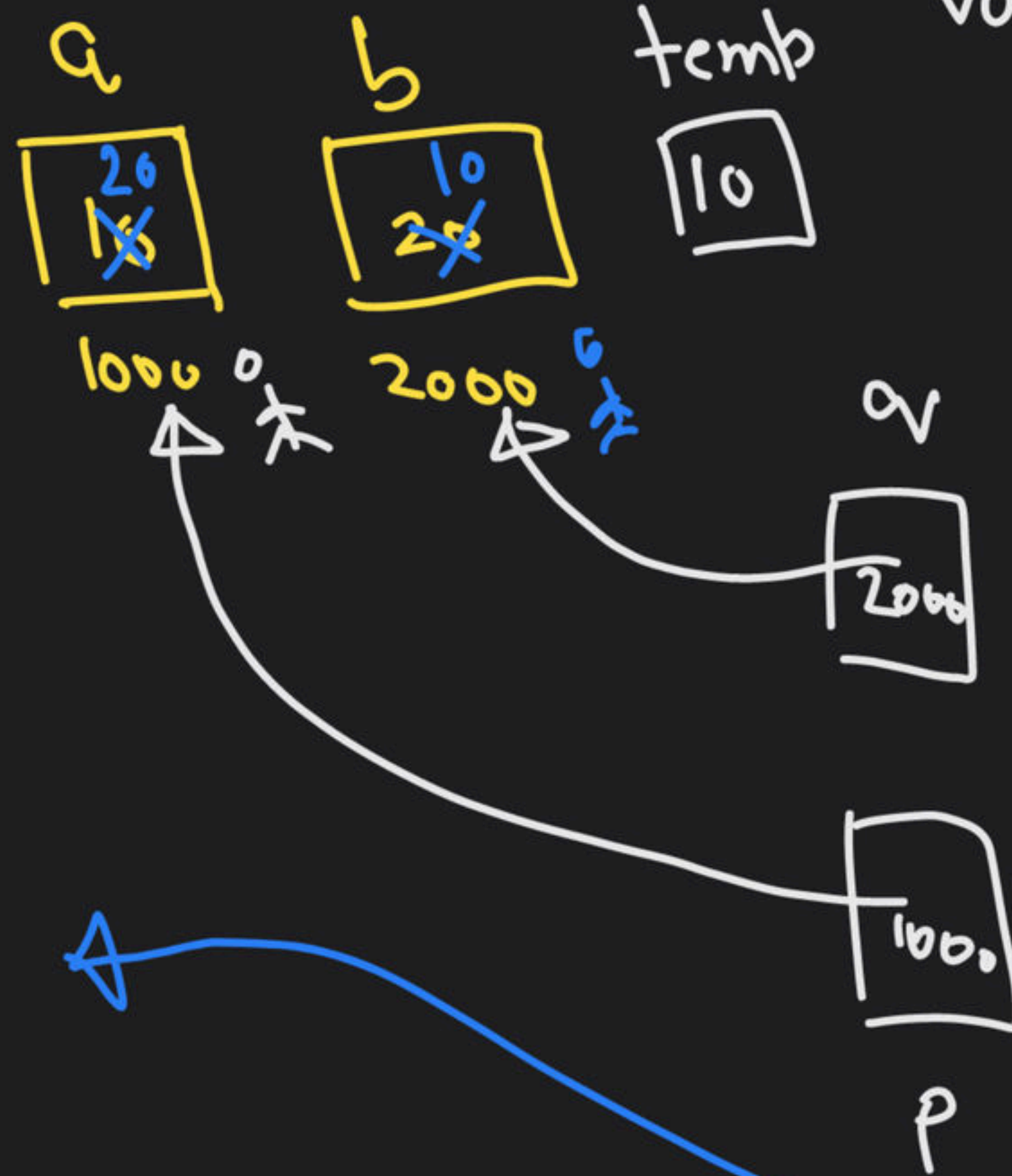
$$* \left(\# + (* q) \right)$$
$$\begin{aligned} (i) \quad *a_v &= *a_{v+1} \\ P[2] &= \cancel{P[2]} + 1 \\ &= a[2] + 1 \\ &= a[3] \end{aligned}$$

(i) use $\text{pf}("1.a", **a)$




```

void swap(int*, int*)
void main() {
    int a = 10, b = 20;
    swap(&a, &b);
    printf("%d %d", a, b);
}
    
```



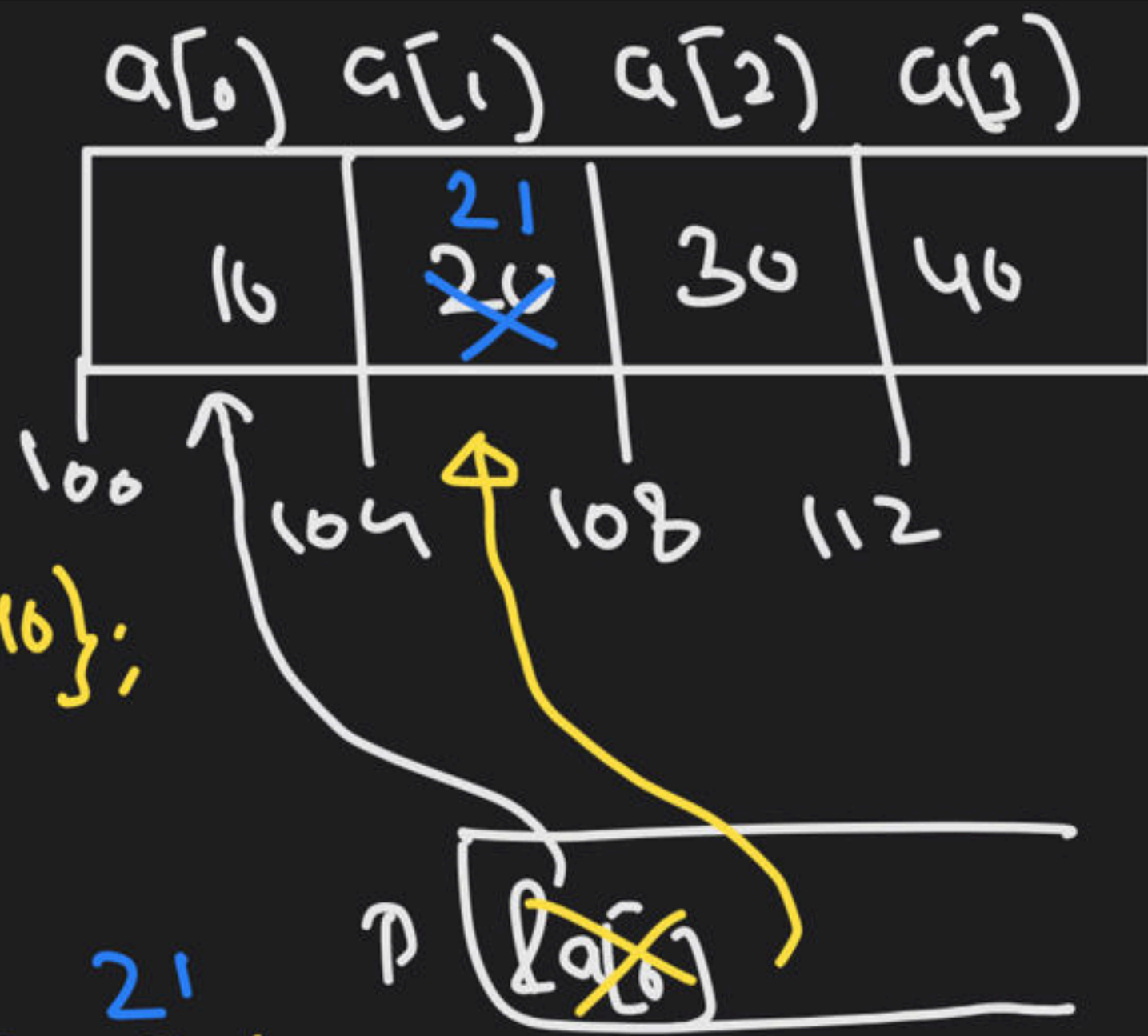
```

void swap(int* p, int* q)
{
    int temp;
    temp = *p;
    *p = *q;
    *q = temp;
}
    
```


unacademy
 void fun(int *);
 void main {

int a[4] = {10, 20, 30, 40};
 fun(a);

printf("%.1d %.1d", a[0], a[1]);
 }

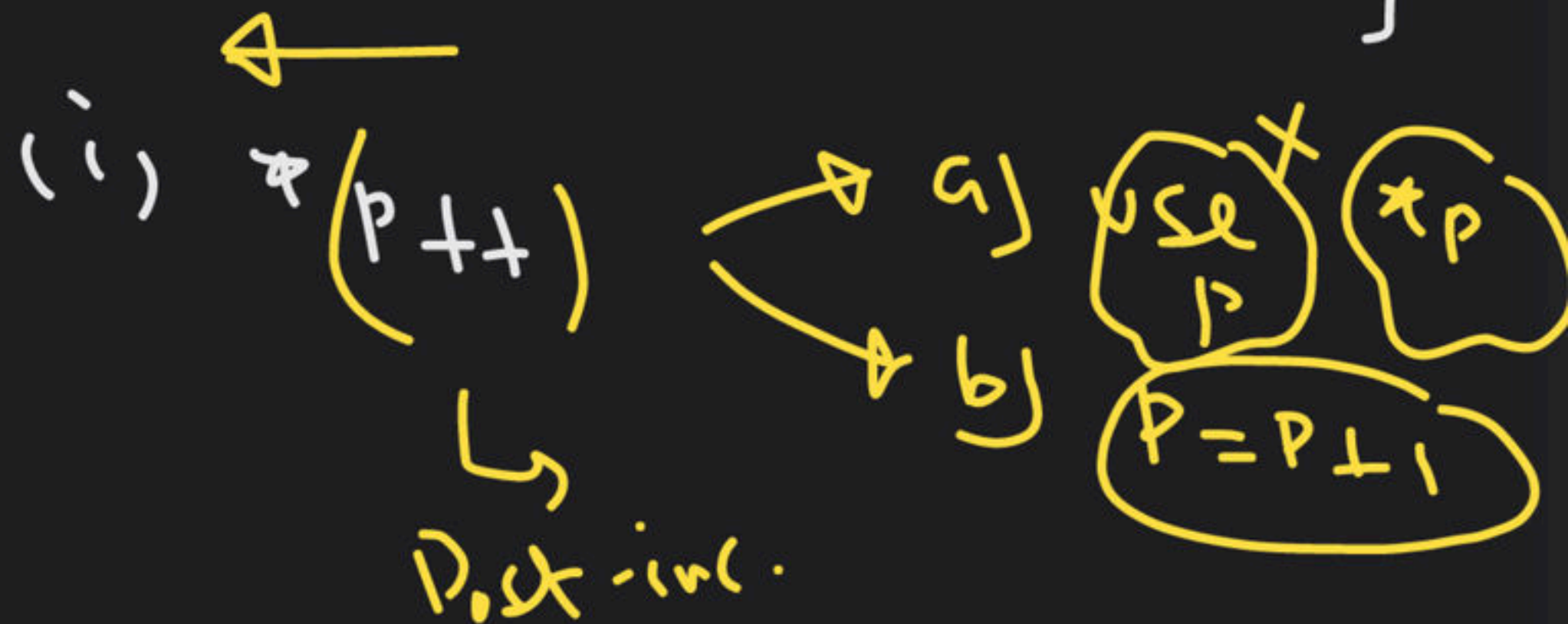


void fun(int *p)
 {
 ✓ ++p;
 ++*p;

}
 ++(*p) a[i] * p = *p + 1
 [i] use X

```
void fun(int*);  
void main() {  
    int a[4] = {10, 20, 30, 40};  
    fun(a);  
    printf(".d.d", a[0], a[1]);  
}
```

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




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

```
void main() {
```

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

```
    fun(a);
```

```
    printf("%d %d", a[1], a[2]);
```

```
    } 20 30
```

```
void fun(int *p)
```

```
{
```

```
    ++*p++;
```

```
}
```

```
++(*p++)
```

```
++*p
```

```
++(*p)
```

a) $*p = *p + 1$ b) $(*p)++$

```
p = p + 1;
```

(i)

(ii)

```
void main{
```

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

```
    la[0]  
    fun(a);
```

|||
Y

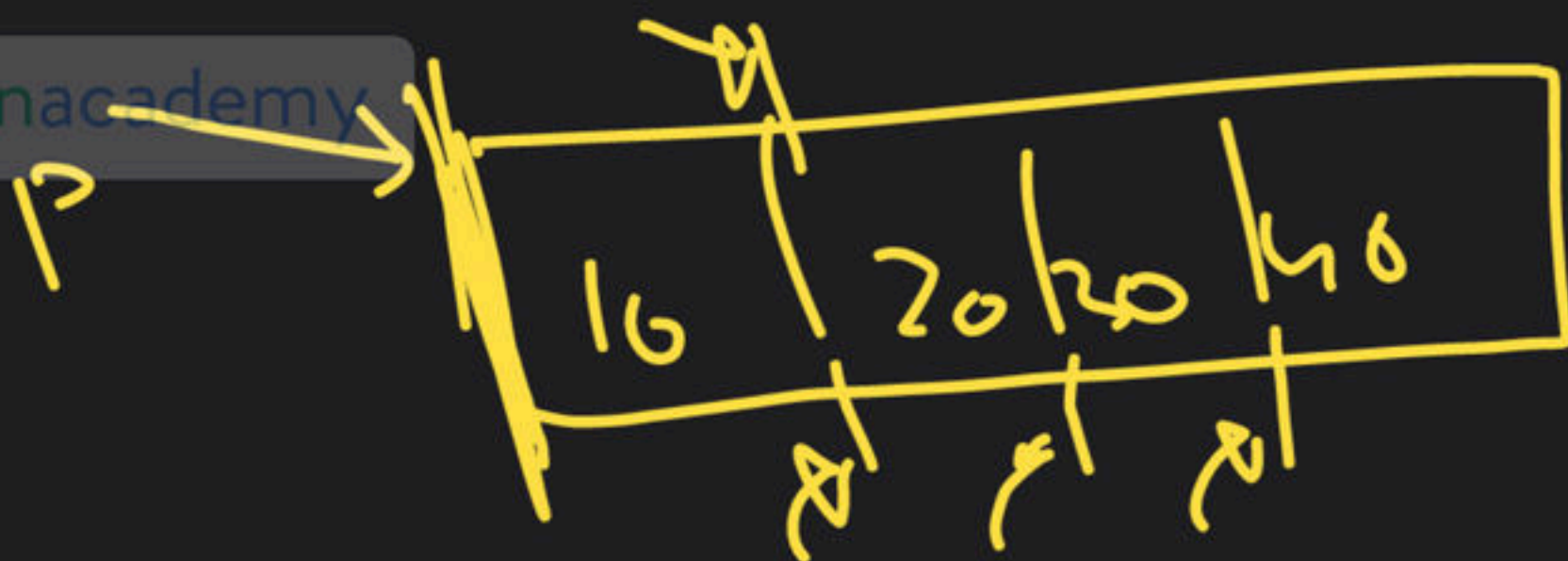
```
void fun(int *p)  
{
```

```
}
```

internally \Rightarrow pointer

```
void fun(int a[])  
{
```

```
}
```

```
void main() {
```

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

```
    sum(a, 4);
```

```
}
```

$\rightarrow (p+0) \rightarrow 10$

$\rightarrow (p+1) \rightarrow 20$

$\rightarrow (p+2) \rightarrow 30$

```
}
```

```
void sum(int *p, int n)
```

```
{
```

```
    int s = 0;
```

```
    for (i = 0; i < 4n; i++)
```

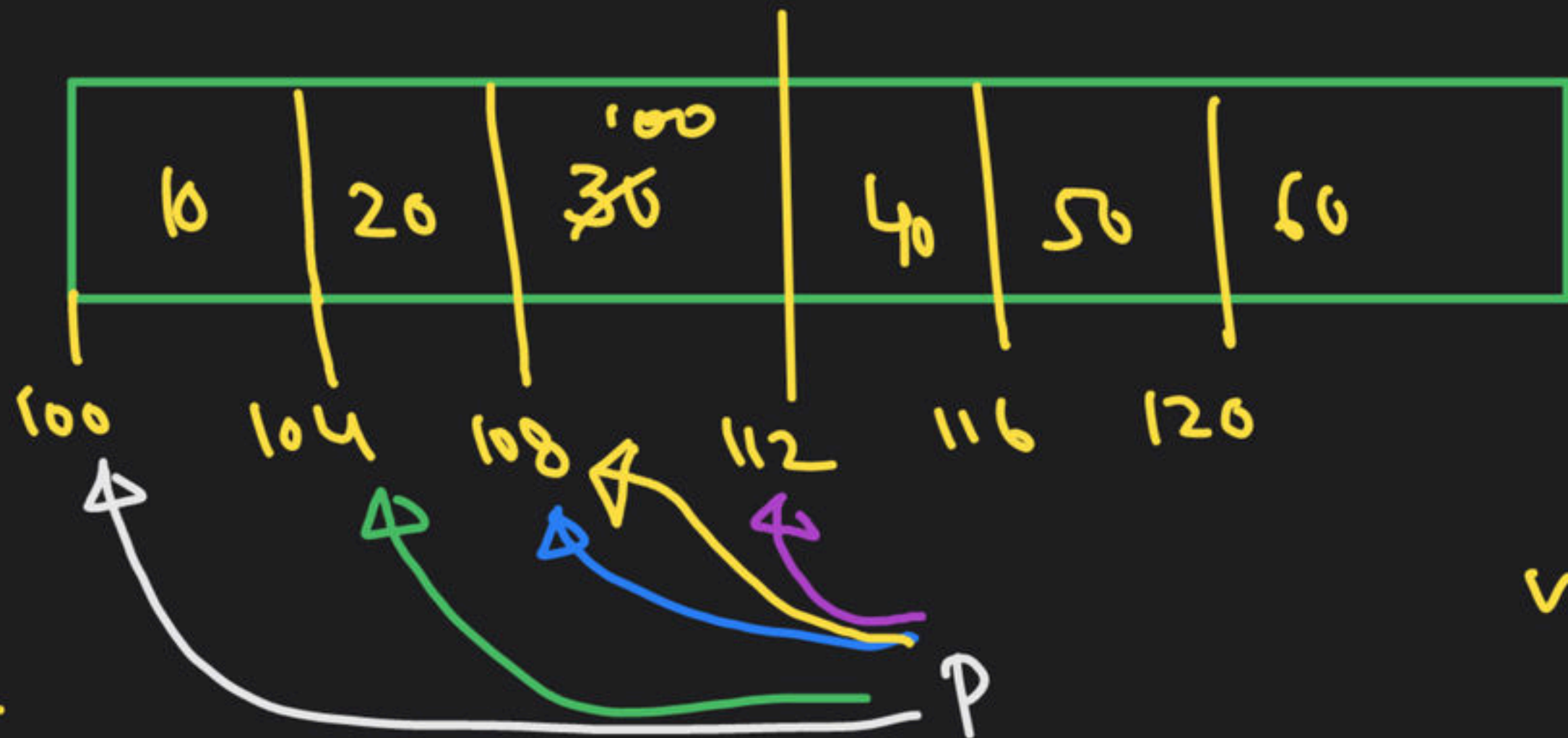
```
{
```

```
        s = s + p[i];
```

```
}
```

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

```
}
```

```
void main() {
```

```
int a[2][3] = {10, 20, 30, 40, 50, 60};
```

```
    fun(a[0]);
```

```
    printf("1. a 1. a 1. a", a[0][0], a[0][1], a[0][2]);
```

```
    }
```

```
void fun(int *P)
```

```
{
```

```
    ++P;
```

```
    *P++;
```

```
    ++P;
```

```
    P--;
```

```
    *P = 100;
```

```
}
```

a) $P = P + 1$
b) $*P$

use $(i++)$ $(i)++$

15 min

```
void main() {  
    int a[2][3] = {10, 20, 30, 40, 50, 60};
```

→ a?
fun(a); → &a[0] ⇒ address of

an array
of
3 integers

```
    printf(".1d .1d .1d", a[0][0], a[0][1],  
           a[0][2]);  
}
```

```
void fun(int (*p)[3])
```

```
{
```

```
    ++p;
```

```
    printf(".1d", *((*p+1)+1));  
};
```


2
Test series

→ ① is

more push → ✓✓

Doubt ?

→

Blueprint



50

50

50

50

200



THANK YOU!

Here's to a cracking journey ahead!