

CS & IT ENGINEERING

Programming in C

Arrays and Pointers 04
Lec- 04



By- Pankaj Sharma sir

TOPICS TO BE COVERED

Pointers

int $x = 10;$

int $\star P;$

int $\star \star q;$

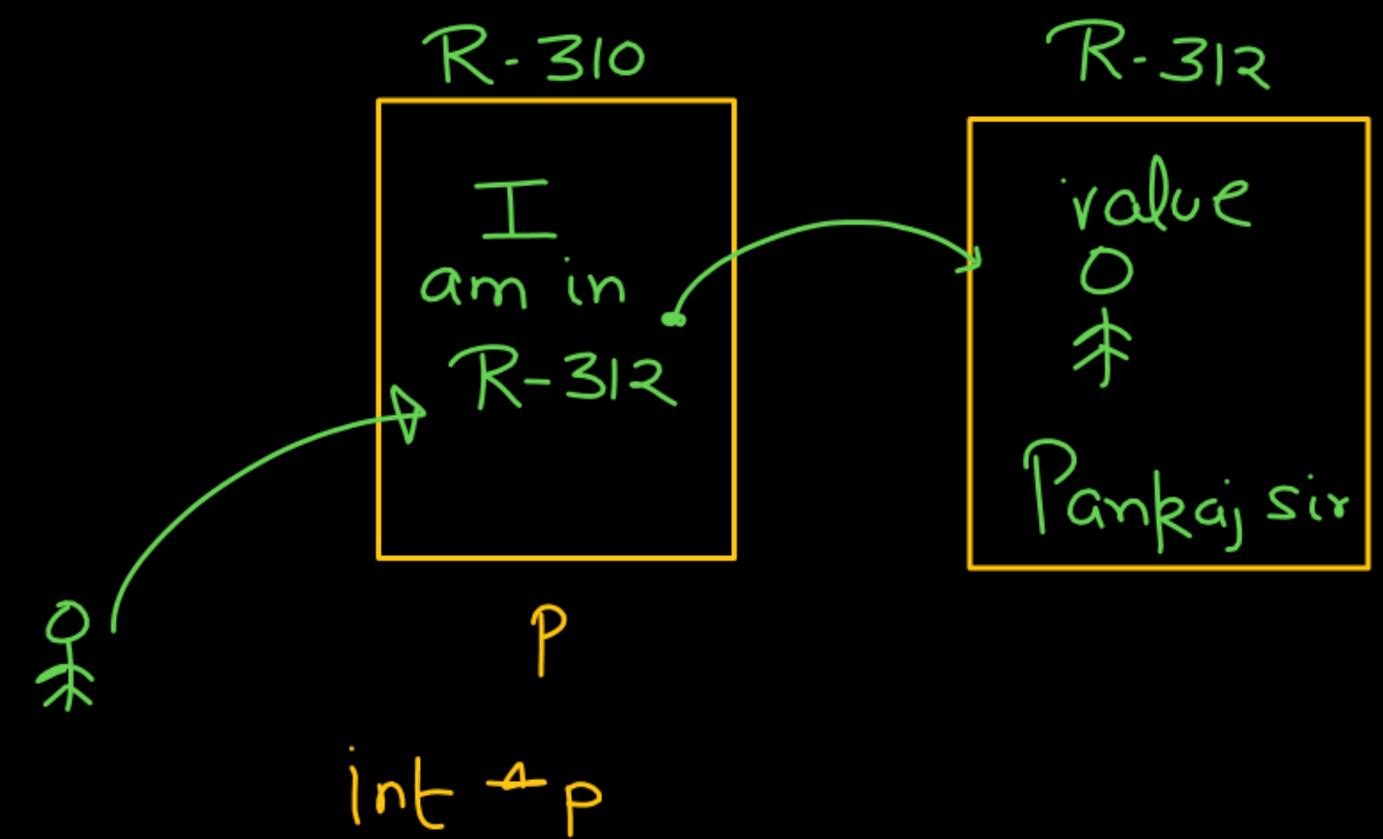
$P \Rightarrow$ address of integer variable

P is a pointer to integer

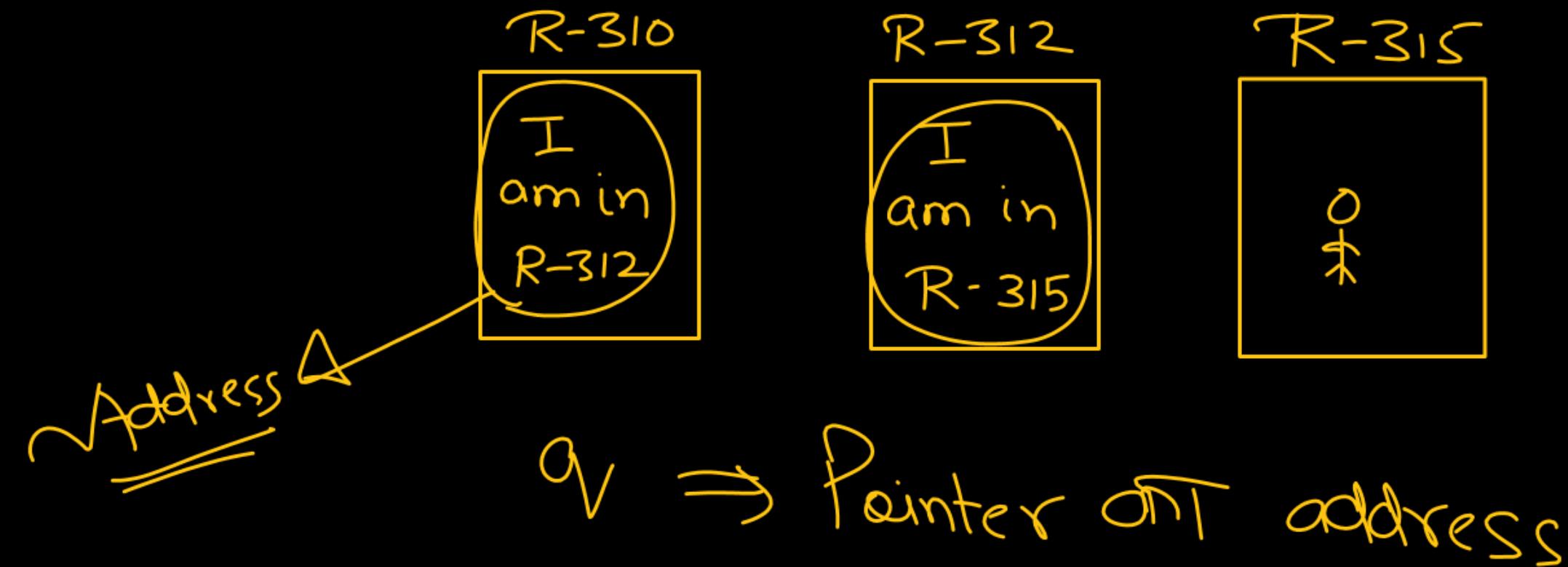
q is a pointer to pointer to integer.

int x=10





$\text{int } *qV$



```
int x = 10;
```

```
int *P;
```

```
int **q; q is a pointer to
```

pointer to int

$*q_v$ = Memory location 1036

$*(*q_v) \Rightarrow$ value at (Memory loc. 1036)
 $\Rightarrow 10$



int $x = 10;$
int $\star P;$

int $\star \star q_v;$
 $P = \&x; \checkmark$

$q_v = \&P;$

1036 printf("%u", P);

✓ 10 printf("%u", *P);

2016 printf("%u", q_v);

1036 printf("%u", *q_v);

10 printf("%u", **q_v);

P = Memory location 1036

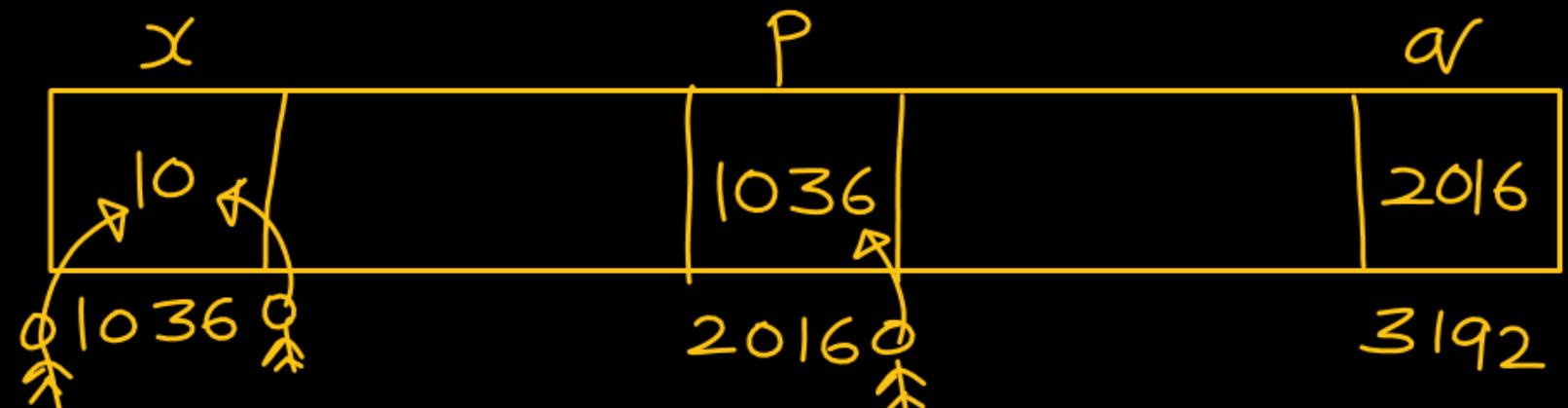
$\star(P) =$ value at (Memory location 1036)
 $\Rightarrow 10$

q_v = Memory location 2016

$\star q_v =$ value at (Memory location 2016)

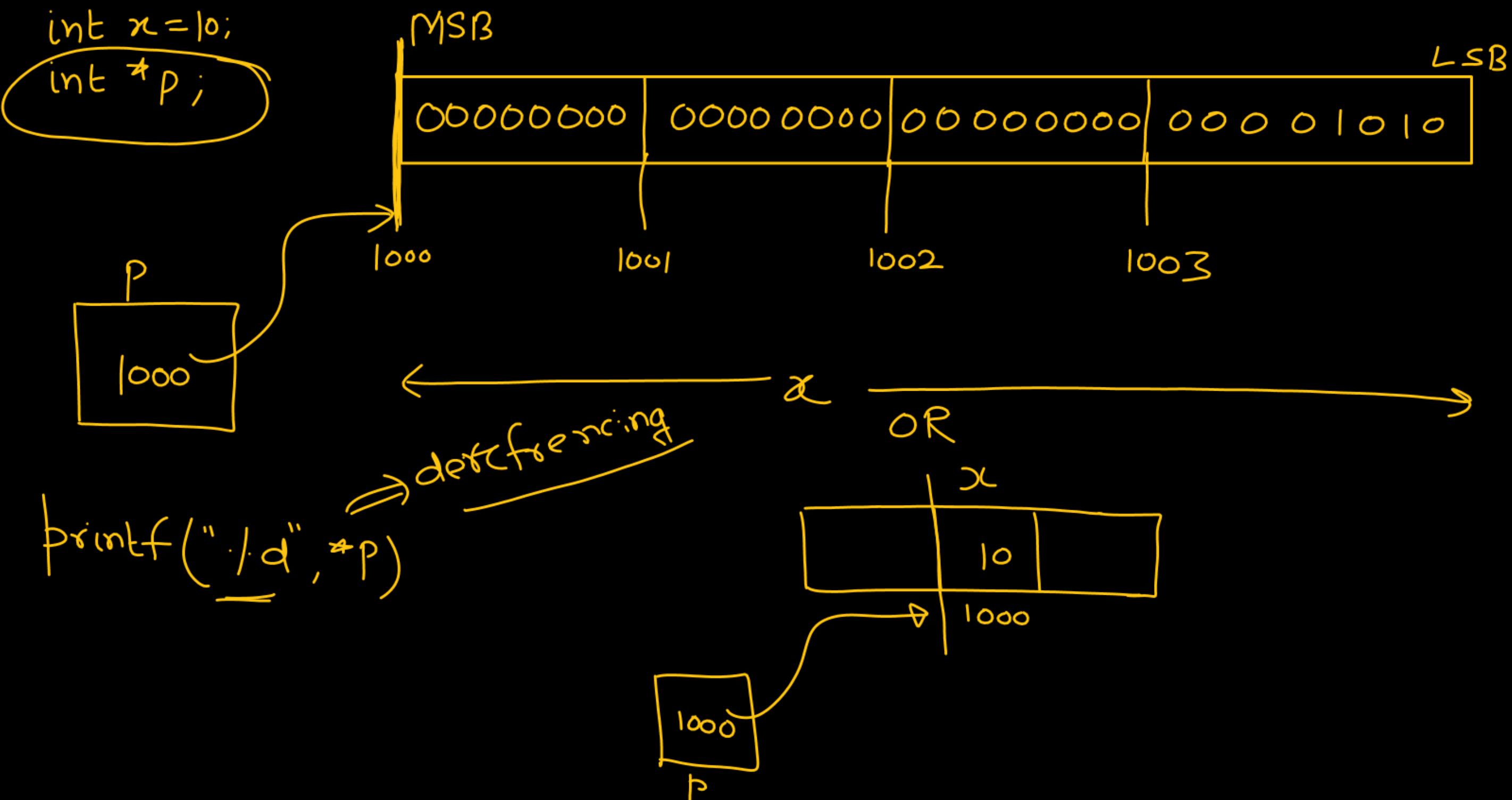
$\star q_v = 1036$
but
1036 is a
memory
location

int $x = 10;$
 int $\star P;$
 int $\star \star q;$
 $P = \&x; \checkmark$
 $q = \&P;$
 1036 $\text{printf}("%u", P);$
 ✓ 10 $\text{printf}("%u", \star P);$
 2016 $\text{printf}("%u", q); \checkmark$
 1036 $\text{printf}("%u", \star q);$
 ✓ 10 $\text{printf}("%u", \star \star q);$



$P = \&x$ $\star P = \star \&x$ $\star P \Rightarrow x$	$q = \&P;$ $\star q \Rightarrow \star \&P$ $\star q \Rightarrow P$ $\star \star q \Rightarrow \&x$ $\star \star q \Rightarrow \star \&x$ $\star \star q \Rightarrow x$
---	---

int x=10;
int *p;



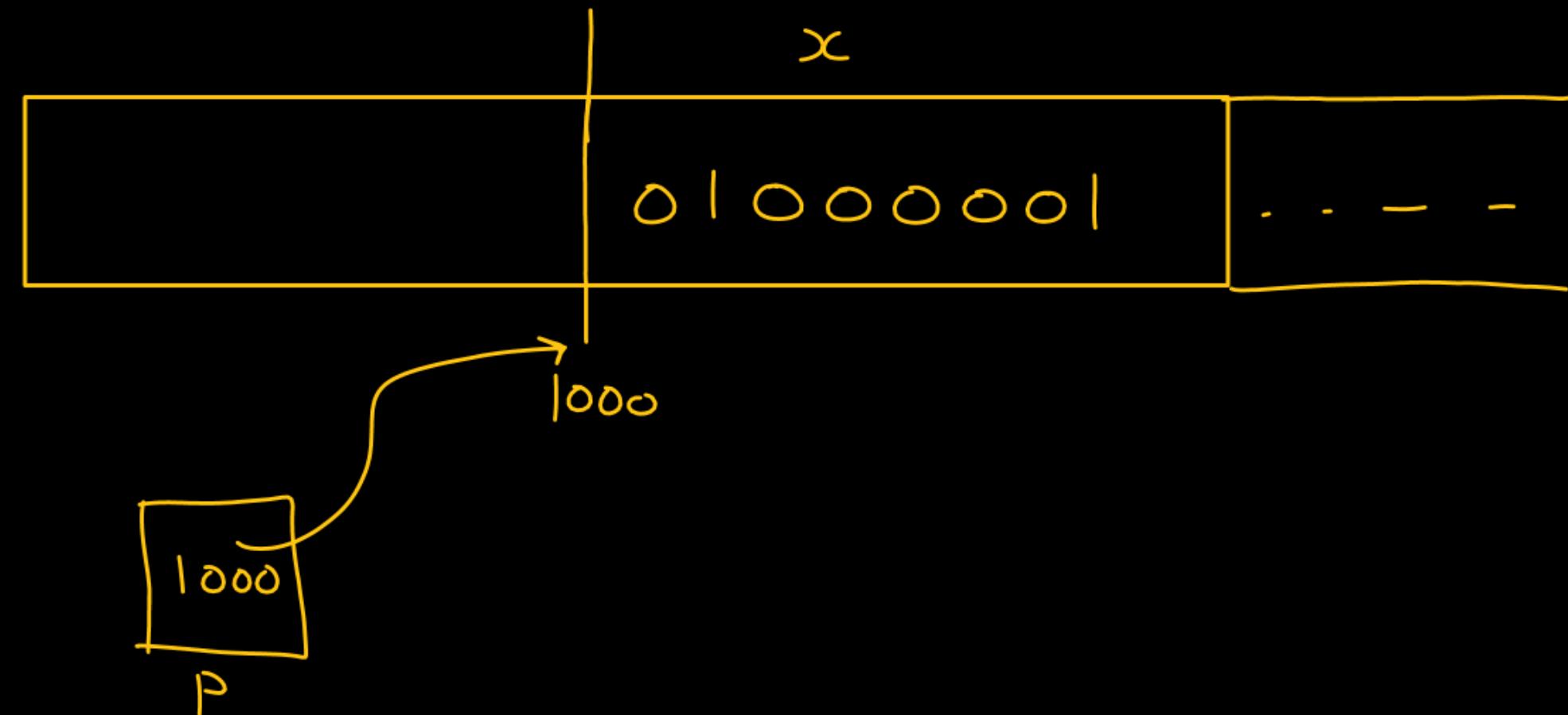
2. $\text{char } x = 65;$

$\text{char } *P;$ P is a pointer to char .

$P = \&x;$

$\text{printf}("%d", *P)$

1 byte



`int *p ;` $p \Rightarrow$ address of int variable

$\text{add} + \text{value} \Rightarrow \text{add}$.

$\text{value} + \text{add} \Rightarrow \text{add}$.

variable

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

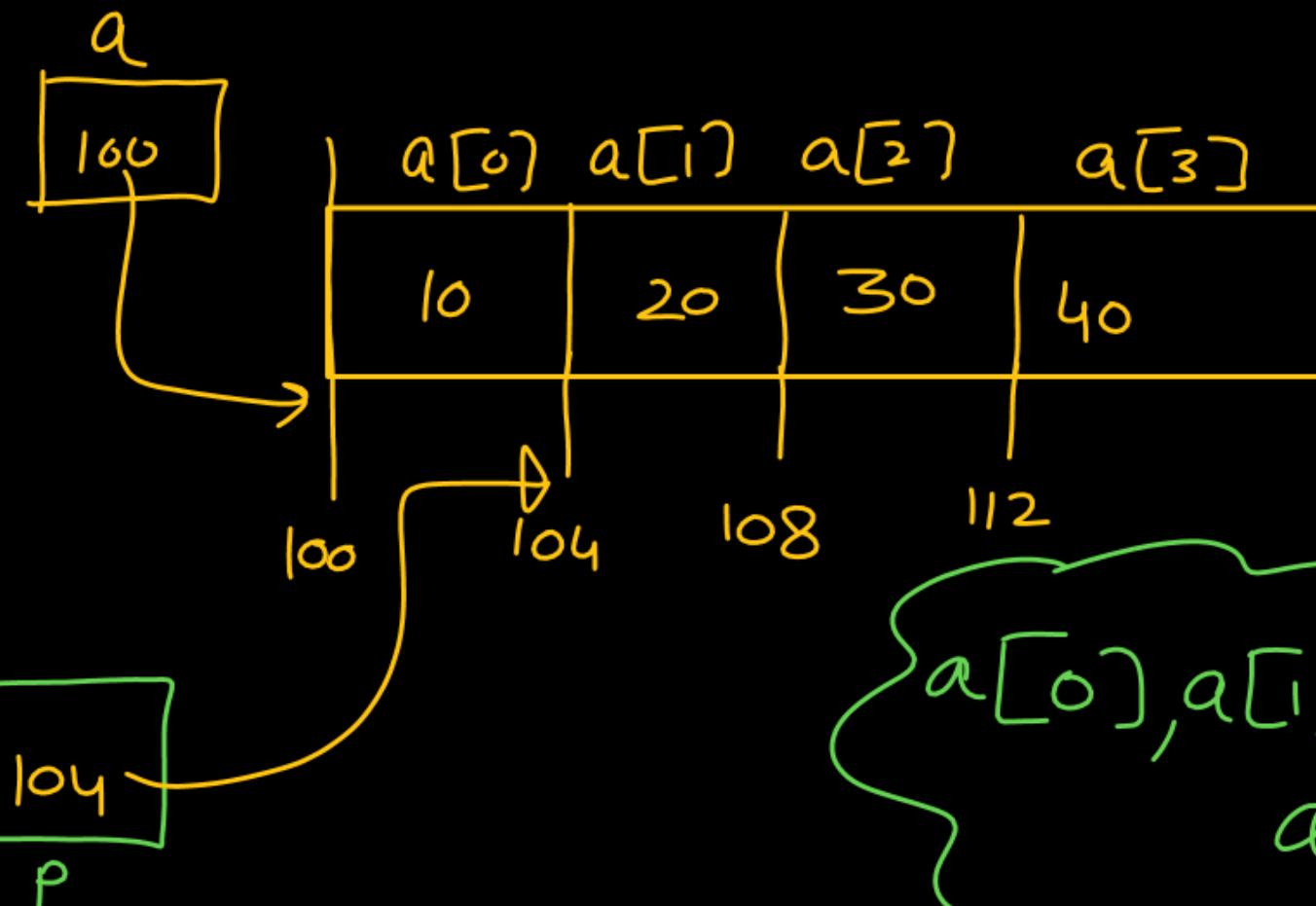
int *P;

P = &a[0]; valid

P = P + 1 ;

Add + value

$$100 + 1 \times 4 \\ P = 104$$



{Add}

(i) ફિરસ્તી add. એ

integer variable

$a[0], a[1], a[2]$
 $a[3]$

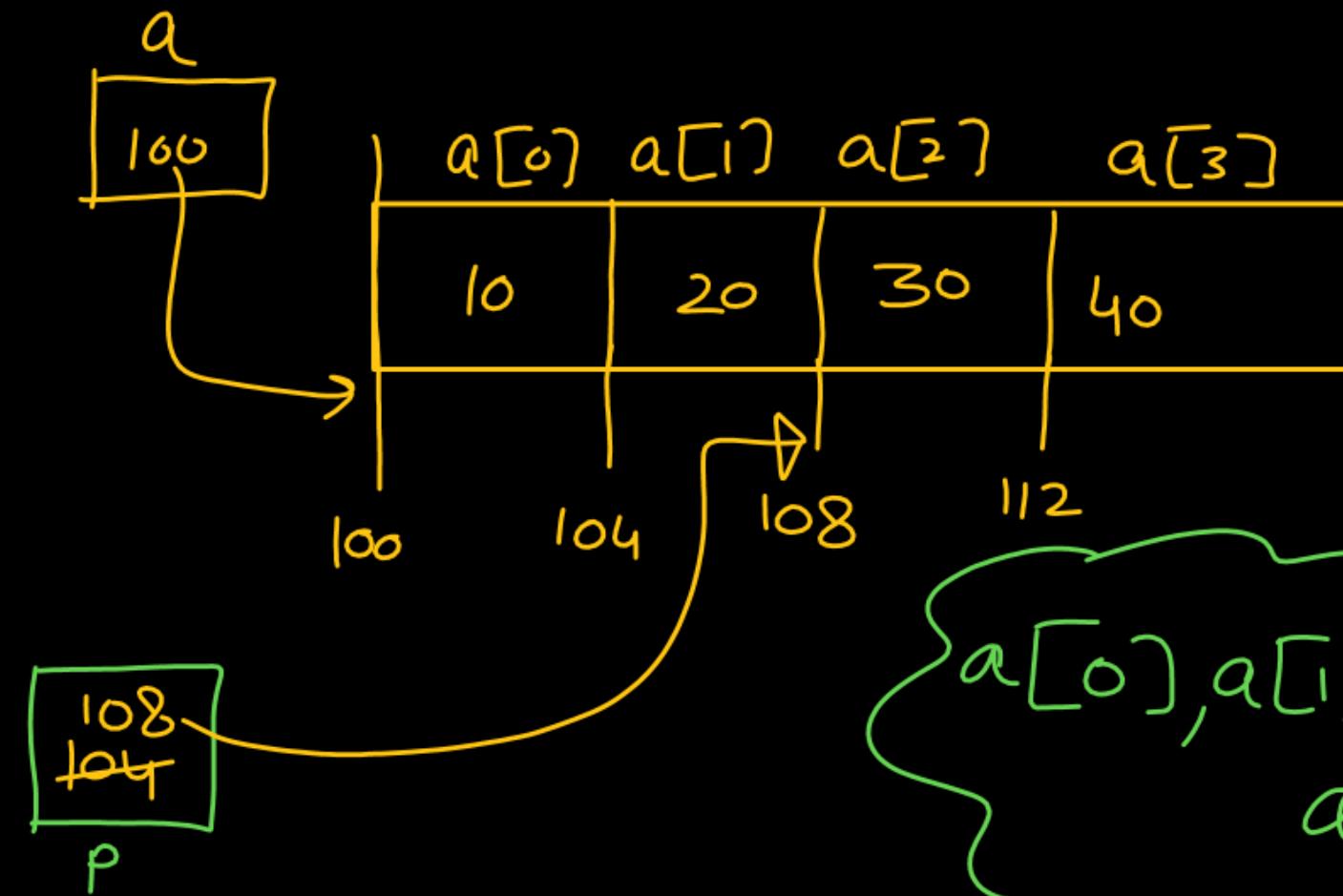
variable

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

P = &a[0]; valid

P++; ✓

P++; P = P+1



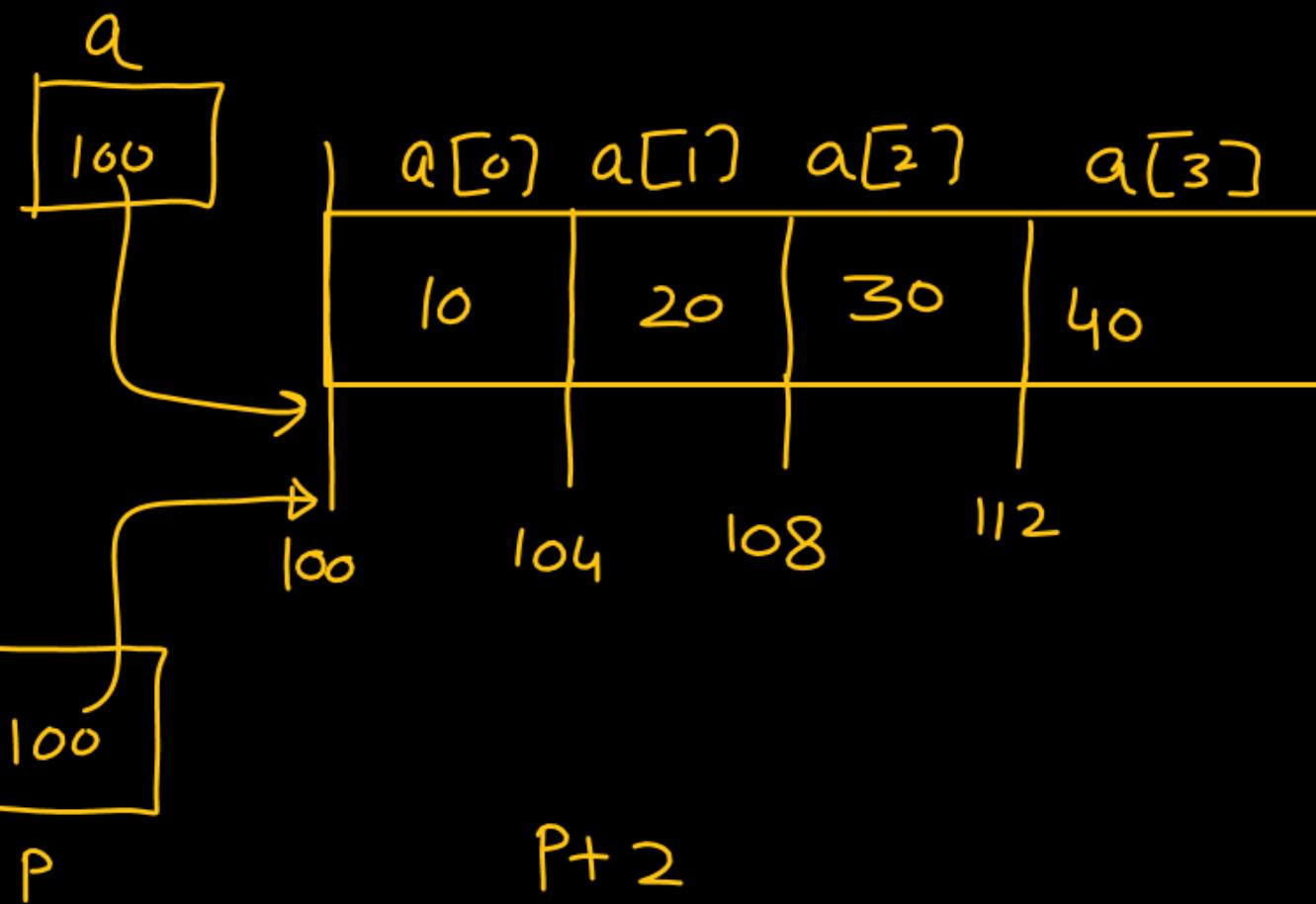
integer variable

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

`[int *P = a;]` valid

OR

`[int *P;
P = a]` valid



(P+2) ? 108

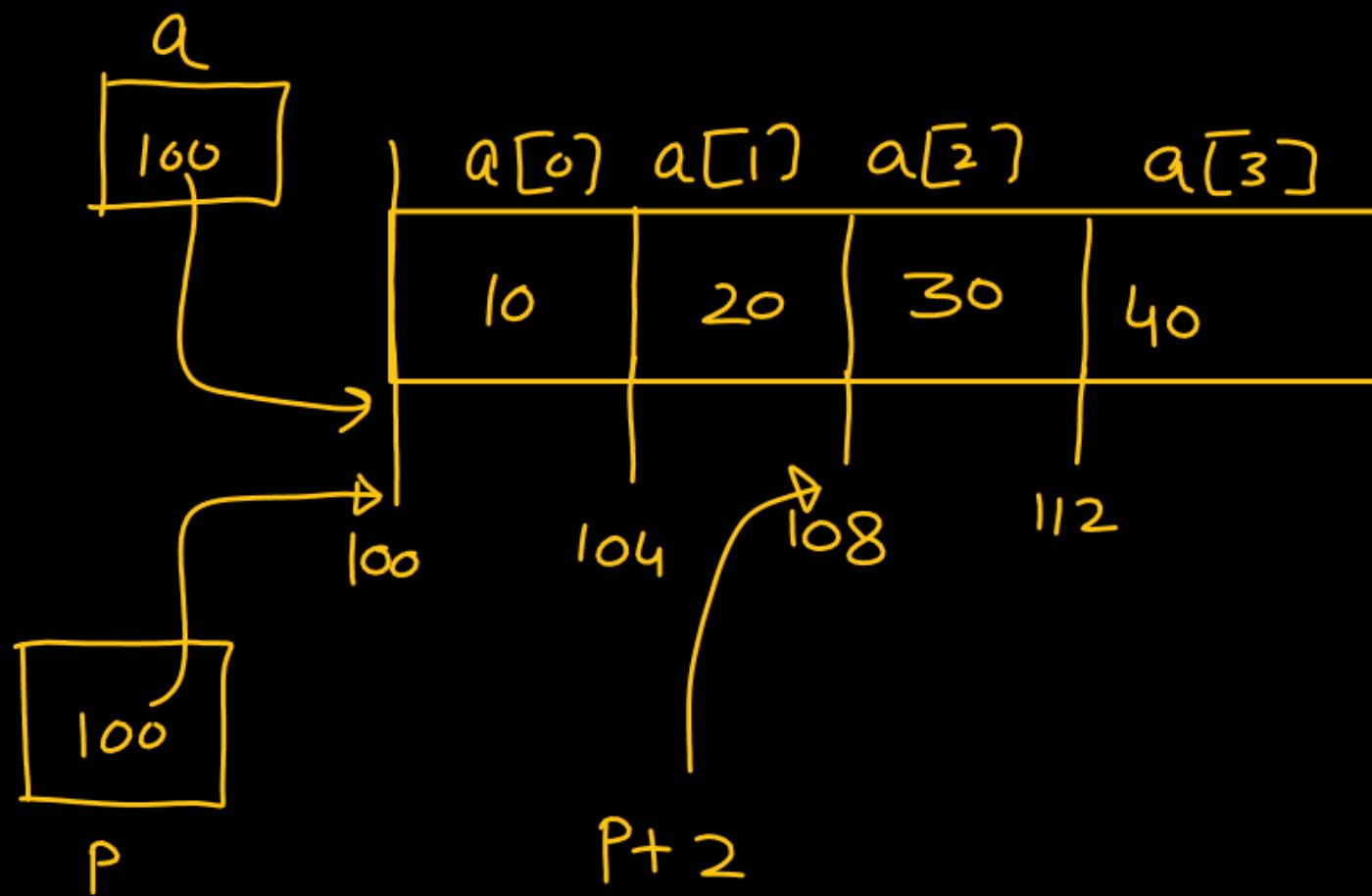
P+3 ? 112

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

```
int *p;
```

① $p + 2$; \times

② $p = (p + 2)$; update



```
void main() {  
    int a;  
        12;  
    a = 12;  
}
```

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

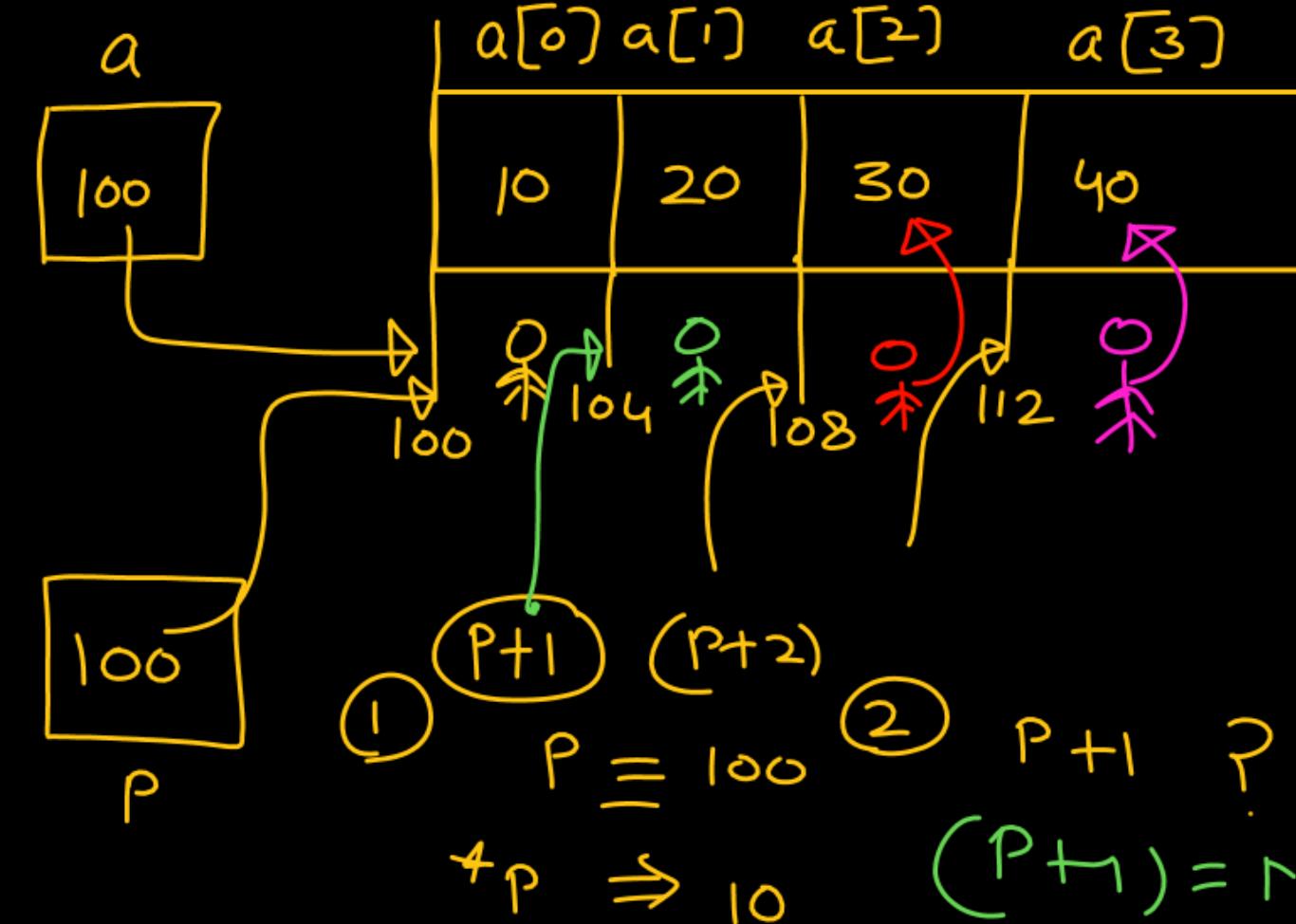
```
int *P = a;
```

```
printf("%d", *P); 10
```

```
printf("%d", *(P+1)); 20
```

```
printf("%d", *(P+2)); 30
```

```
printf("%d", *(P+3)); 40
```



④ $(P+3) \Rightarrow \text{Mem. location}$
 $\star(P+3) = \text{value at} (\text{Mem. loc.})$
 $\star(P+3) = 40$

③ $P+2 \ ?$
 $P+2 = \text{Memory loc.}$

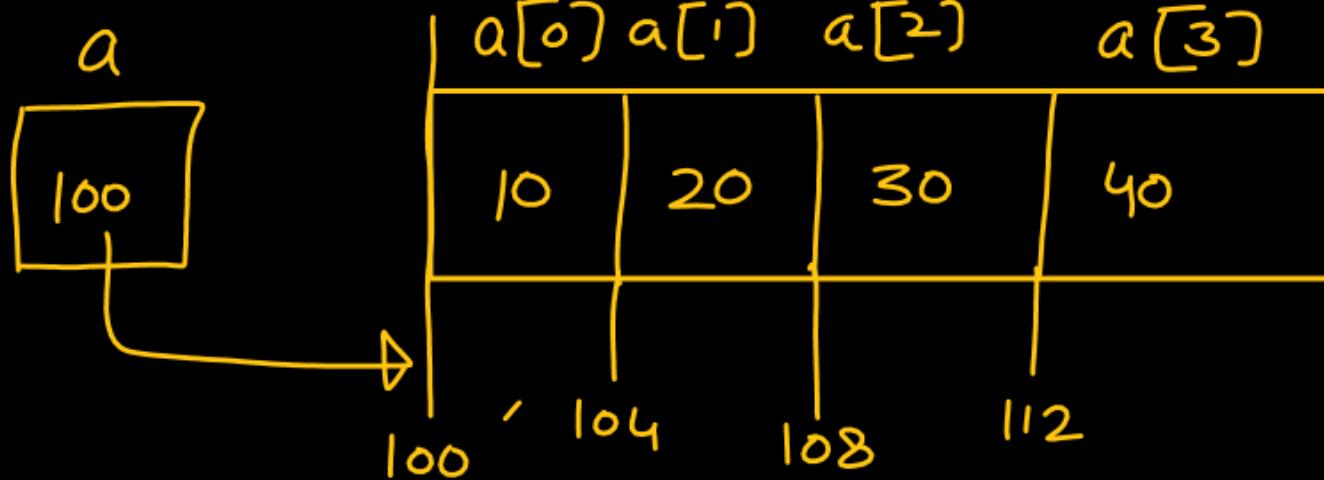
$\star(P+2) = \text{value at} (\text{Memory loc.})$
 $\star(P+2) = 30$

$\star(P+1) = \text{value at} (\text{Mem. loc.})$
 $\star(P+1) = 20$

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

```
int *P = a;
```

```
printf("%d", *P);
printf("%d", *(P+1));
printf("%d", *(P+2));
printf("%d", *(P+3));
```



100
P

$$\star(x+i) = x[i]$$

```
printf("%d", P[0]);
printf("%d", P[1]);
printf("%d", P[2]);
printf("%d", P[3]);
```

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

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

a++;
a--;
--a;
++a;

Invalid

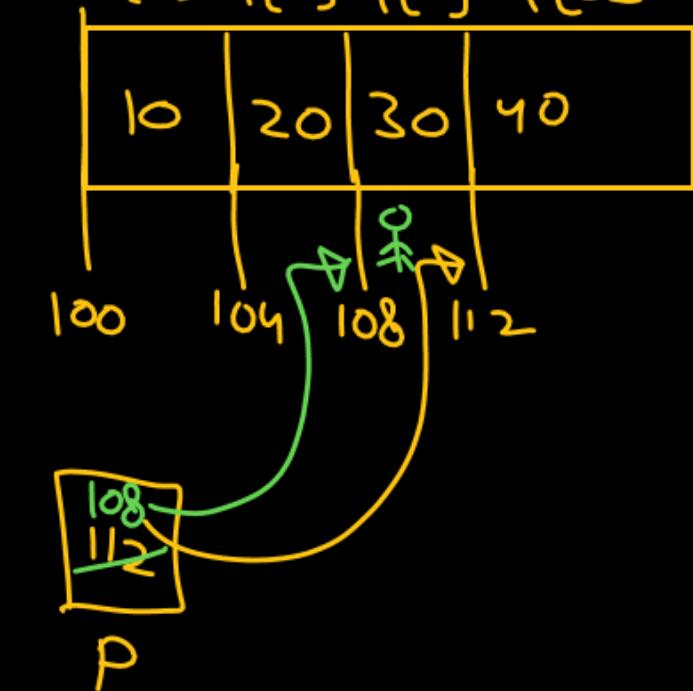
1. int a[4] = {10, 20, 30, 40};

int *P = a;
P++;
P++;

[]

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

int *P = &a[3]; a[0] a[1] a[2] a[3]
--P;
printf("%d", *P);



add + add

① Pointer + Pointer X

② ++ Pointer-variable ✓

-- Pointer-variable ✓

Pointer-variable ++ ✓

Pointer-variable -- ✓

Pointer-variable + 3 \Rightarrow moving 3 positions
in forward direction

Pointer-variable - 3 \Rightarrow moving 3 positions
in backward direction

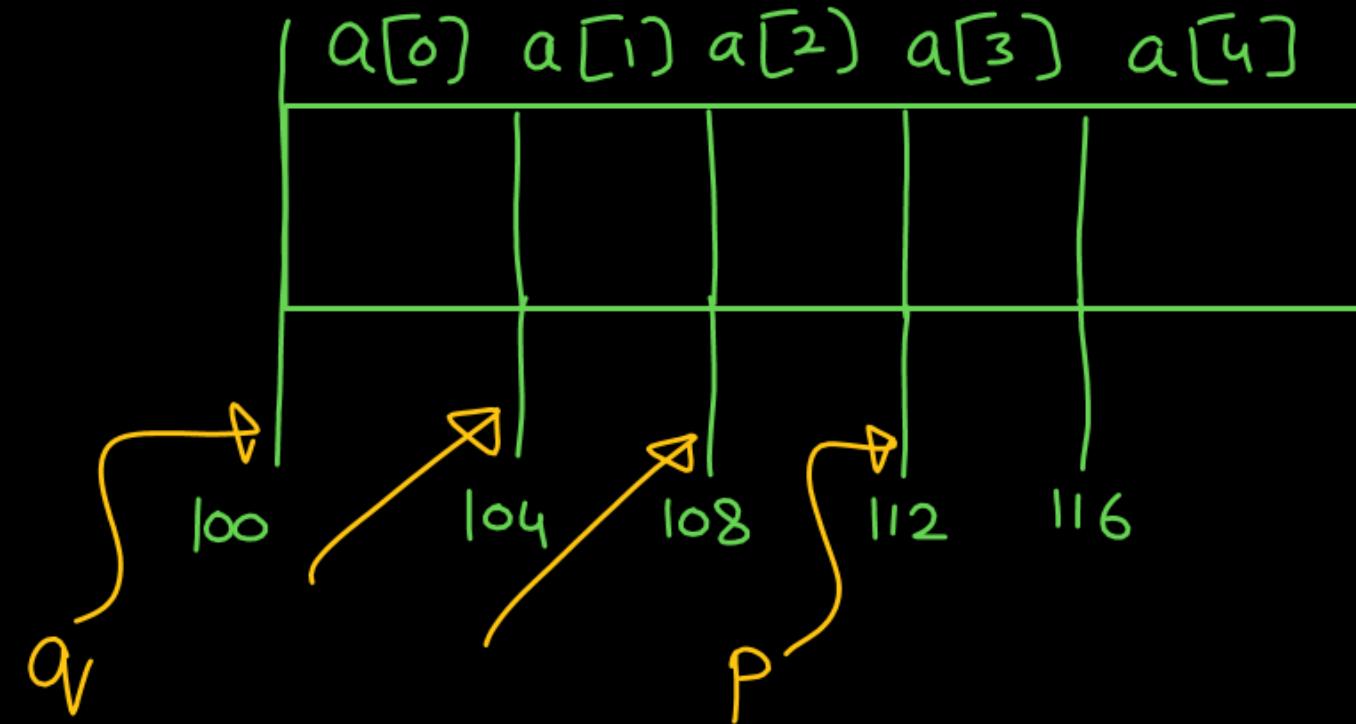
③ Subtraction :

`int *P, *q;`

`P = &a[3]; ✓`

`q = &a[0]; ✓`

$a_1 - a_0 +$



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

Add to diff $\Rightarrow \frac{\text{Actual diff}}{\text{Size of data}}$

④

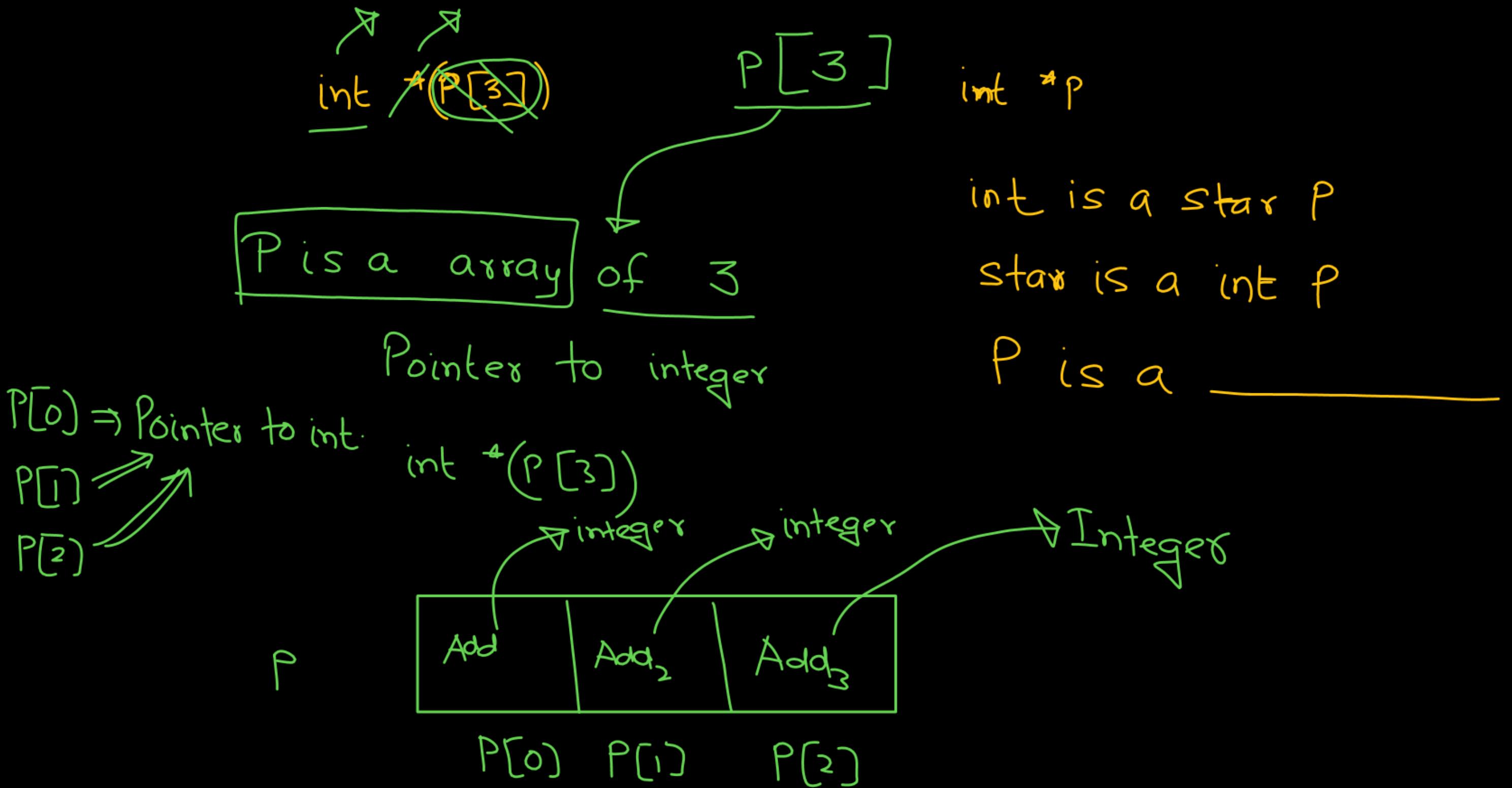
$\text{++}, \text{*}$ $\Rightarrow \text{same}$

R to L

Complex
declaration

```
int *P[4];  
int (*P)[4];  
int *P(int,int);
```

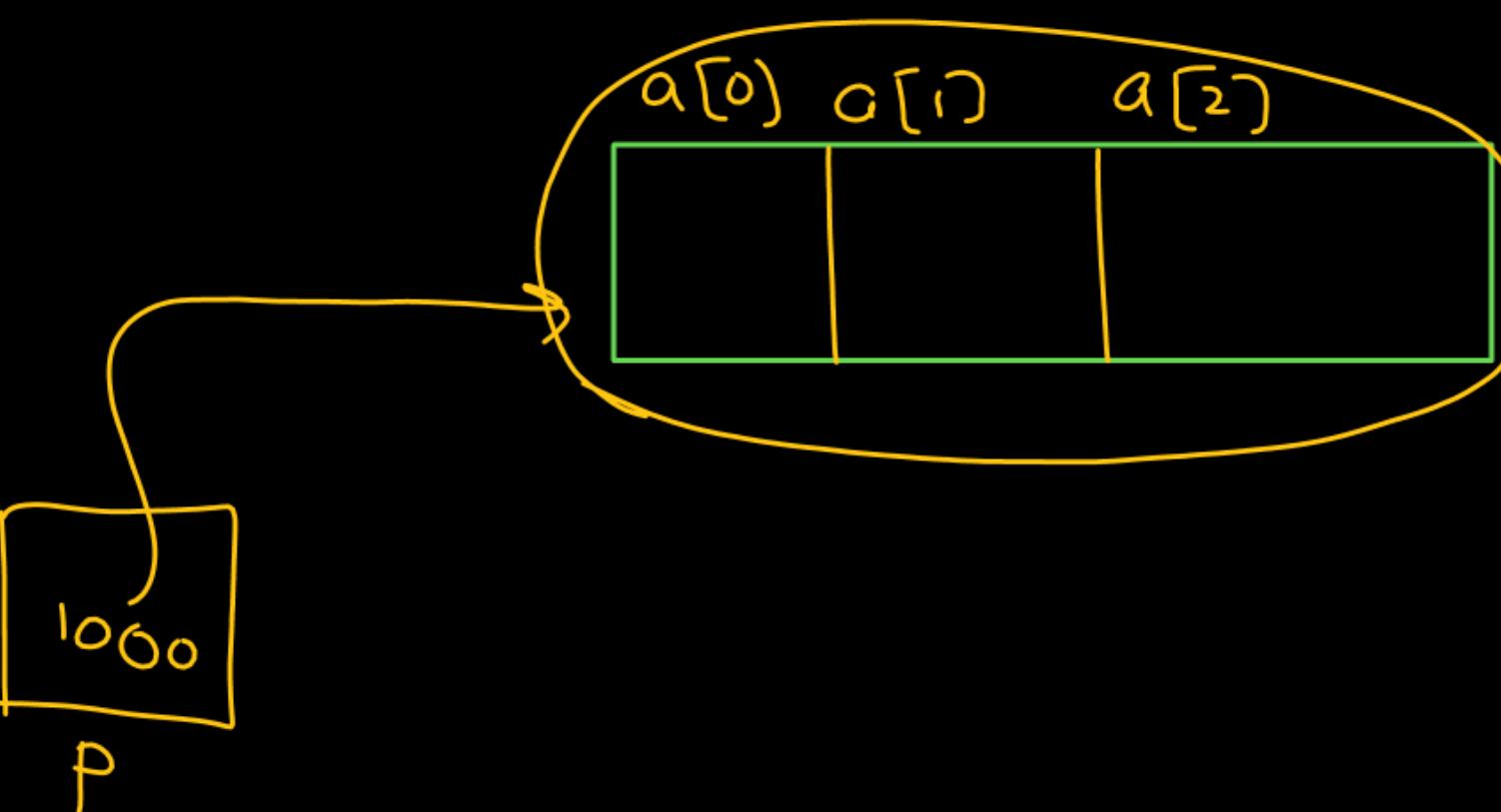
- 1) () functions] ① L to R
- 2) [] Array]
- 3) * Pointer] 2 R to L
- 4) Identifier Var-name, fun-name]
- 5) Data type int, char, float] 3



③ ① ②
int ~~(*P)~~ [3];

P is a pointer to

array of 3 integers



int a [3];

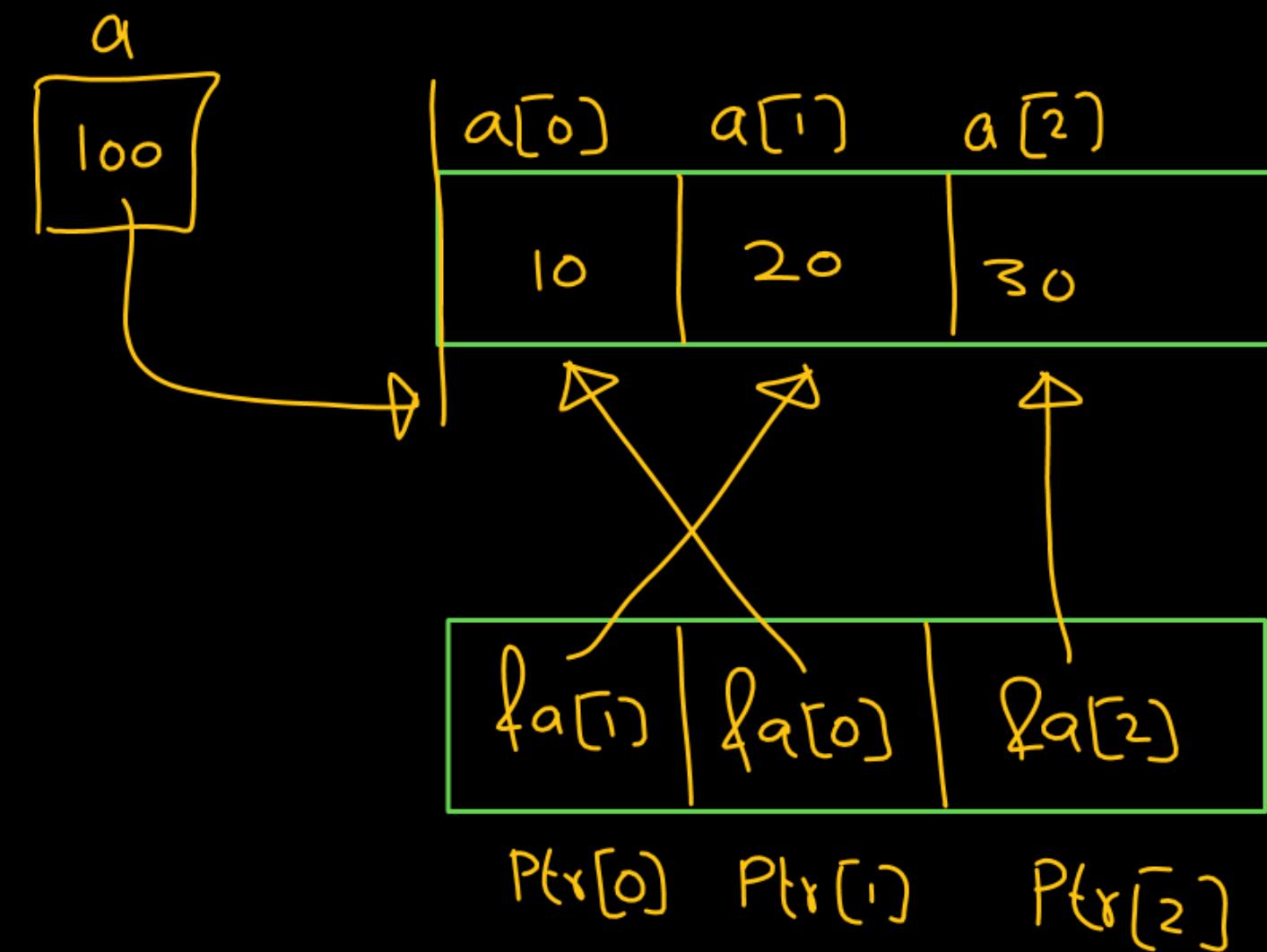
int (*P)[3];

P = &a;

पुरे
अर्रेय
का

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

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



int *a_v ;

$$q = \rho t \gamma;$$

* } same
++ }
R to L

① (++a;)

2

③ $\text{pf}(" / _4; + \left(++ \left(\begin{smallmatrix} 4 \\ 9 \end{smallmatrix} \right) \right))$;

$$(i) \quad *q_V = *q_V + 1$$

$$\nexists \rho_{\text{tr}}[z] = \nexists \rho_{\text{tr}}[z] + 1$$

$$\begin{aligned}P_{\text{Ex}}[z] &= P_{\text{Ex}}[z] + 1 \\&= \delta_0[z] + 1\end{aligned}$$

σ

Expert

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

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

inc ~~on 21~~

$$= q_1 + 1$$

9

a
100

The diagram illustrates a variable *t* pointing to the first element of an array *a*. The array *a* is represented as a horizontal box divided into four cells. The first cell contains *a[0]* with the value 10. The second cell contains *a[1]* with the value 20. The third cell contains *a[2]* with the value 30. The fourth cell contains *a[3]* with the value 40. A yellow bracket labeled *t* points to the leftmost cell, indicating that *t* is assigned the value of *a[0]*.

$f_a[1]$	$f_a[0]$	$f_a[-3]$
$P[x[0]]$	$P[x[1]]$	$P[x[-2]]$

Pt-8

1

1

$$\Pr[2] = f_2[2] + 1^q \\ = f_2[3] + 1$$

$$= 49\lceil 3 \rceil$$

1

1

$$\textcircled{1} \quad q \quad \overline{q}$$

~~→ ↗ APLS[2]~~

```
void swap( int * , int * );
```

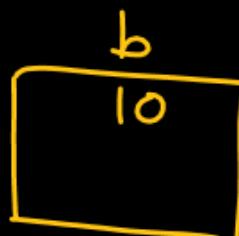
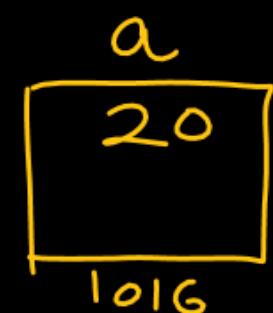
```
void main() {
```

```
    int a = 10, b = 20;
```

```
    swap( &a, &b );
```

```
    printf( "%d %d", a, b );
```

```
}
```



formal arg
void swap(int *x, int *y)

```
{
```

```
    int temp;
```

```
    temp = *x;
```

```
    *x = *y;
```

```
    *y = temp;
```

```
}
```

