

CS & IT ENGINEERING

Data Structures

Introduction to Data Structure
Lec- 06



By- Pankaj Sharma sir

TOPICS TO BE COVERED

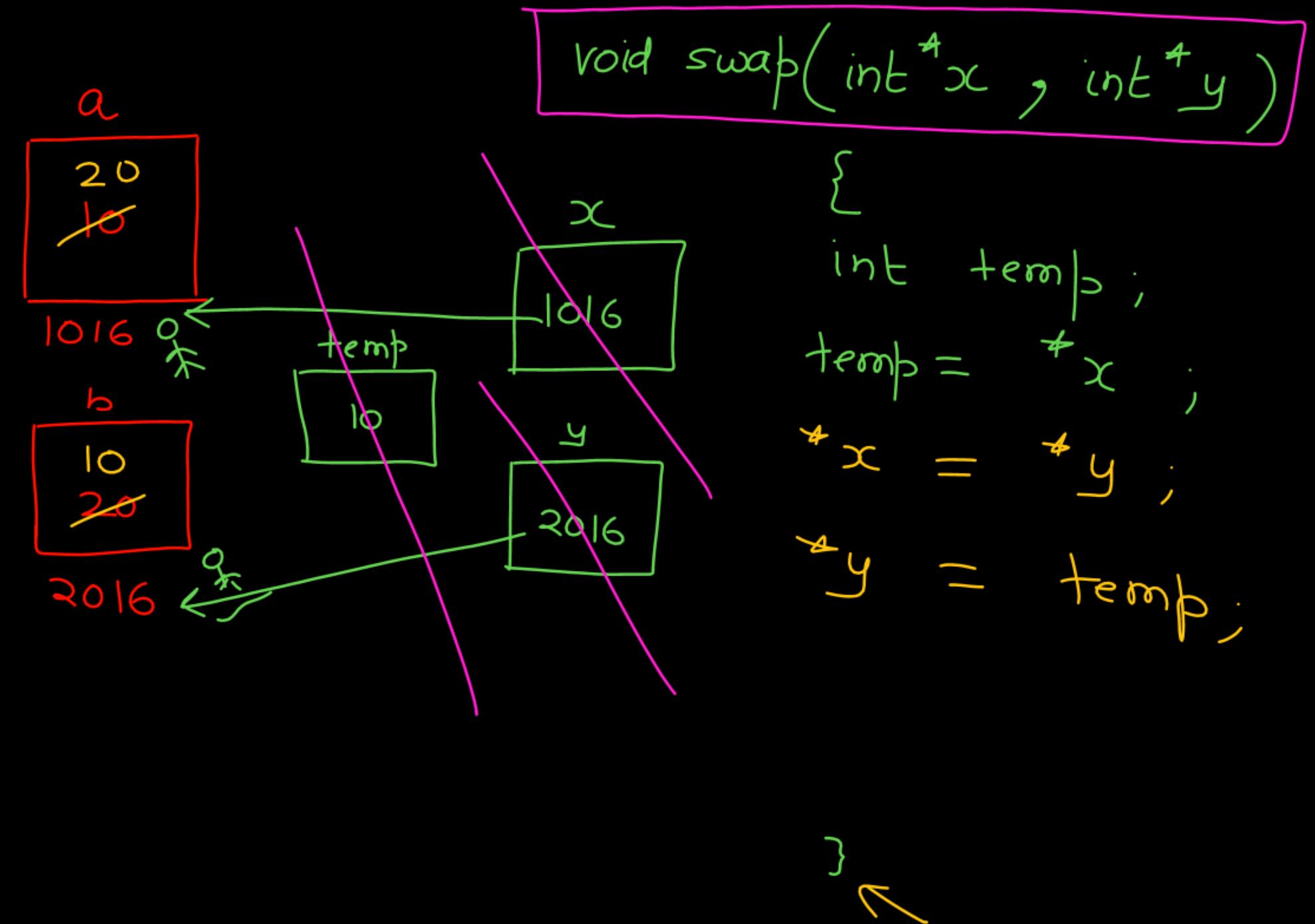
Introduction-6

```

#include<stdio.h>
void swap( int * , int * );
void main( ){
    int a=10,b=20 ;
    printf( "%d %d",a,b);
    swap( &a,&b );
    printf( "%d %d",a,b);
}

```

Call by reference



int $\star(P[4])$;
int $(\star P)[4]$;

declarations?

int $\star P$;

Identifiers

~~1) int is a star P~~

~~2) star is a int P~~

3) P is a pointer to integer

int Add(int,int);

1) ()

Parenthesis

2) []

square brackets

3) Identifiers

name of var, or array,
function

4) *

Pointer

5) Data type

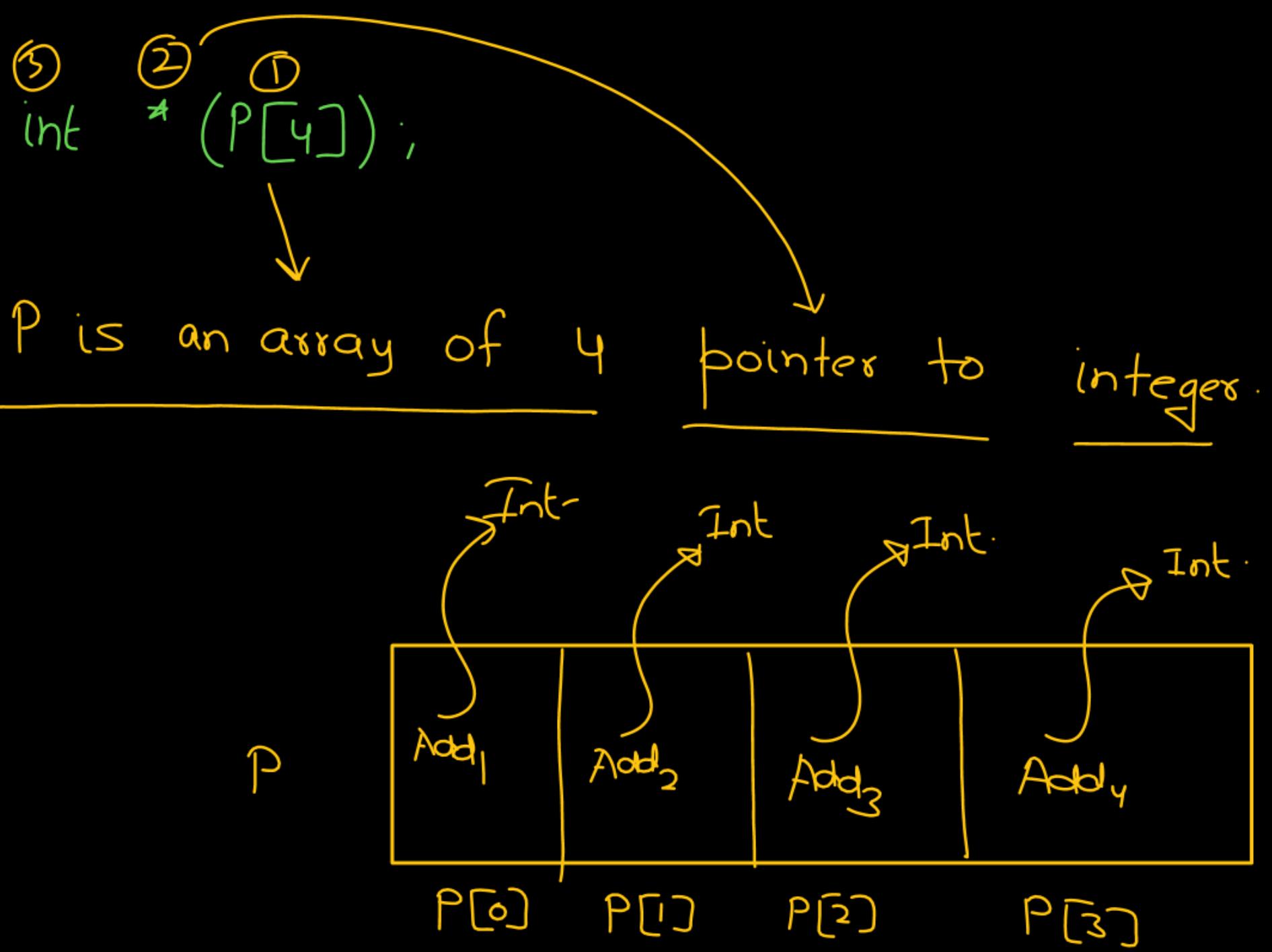
char, int, float, - -

functions
array

Preced
()
L to R

2 R to L

Note: {Mentos idea
Identifier}



$P[0]$
 $P[1]$
 $P[2]$
 $P[3]$

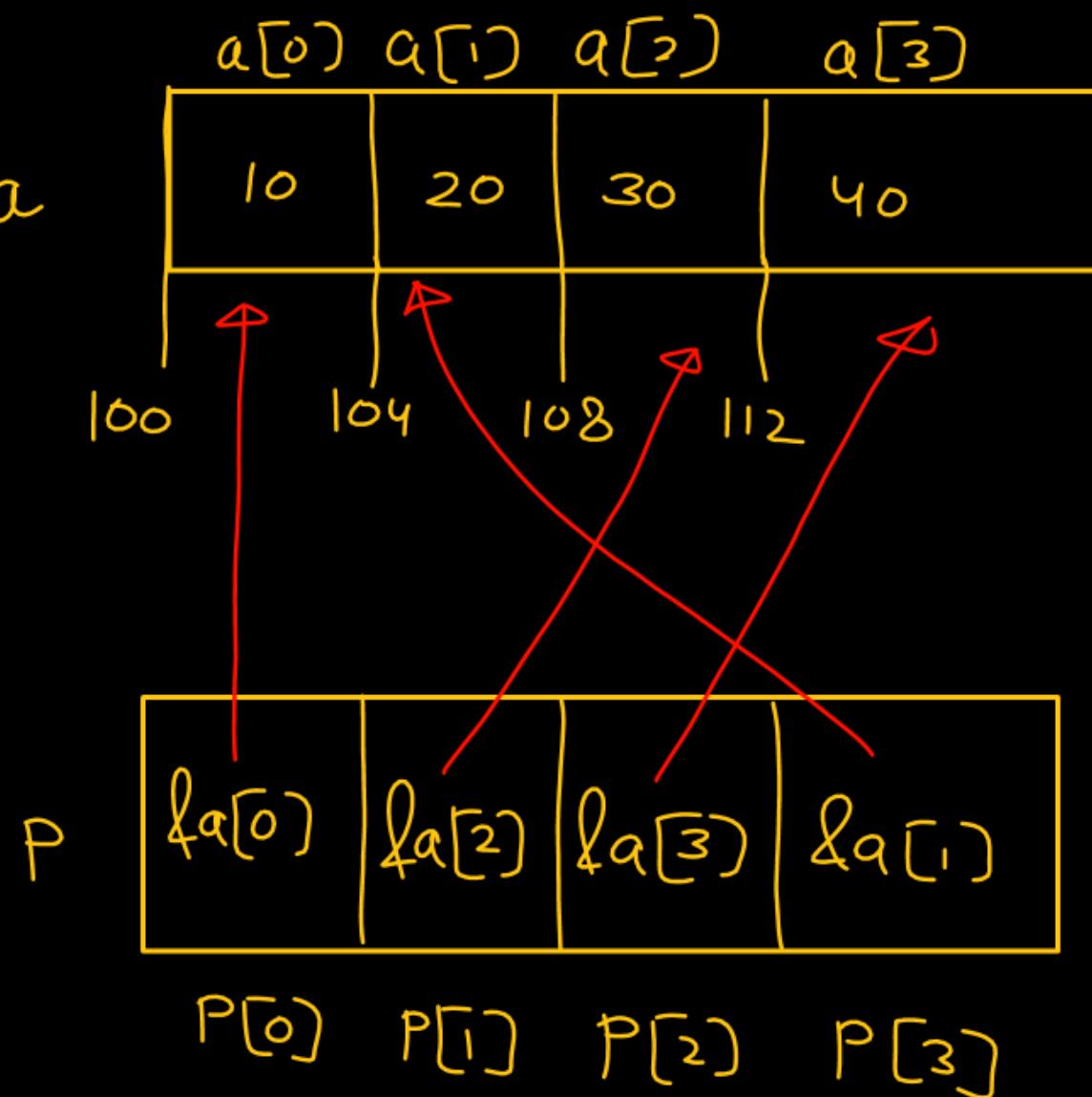
} pointer to integer

$a = \&a[0]$

Ex1.

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

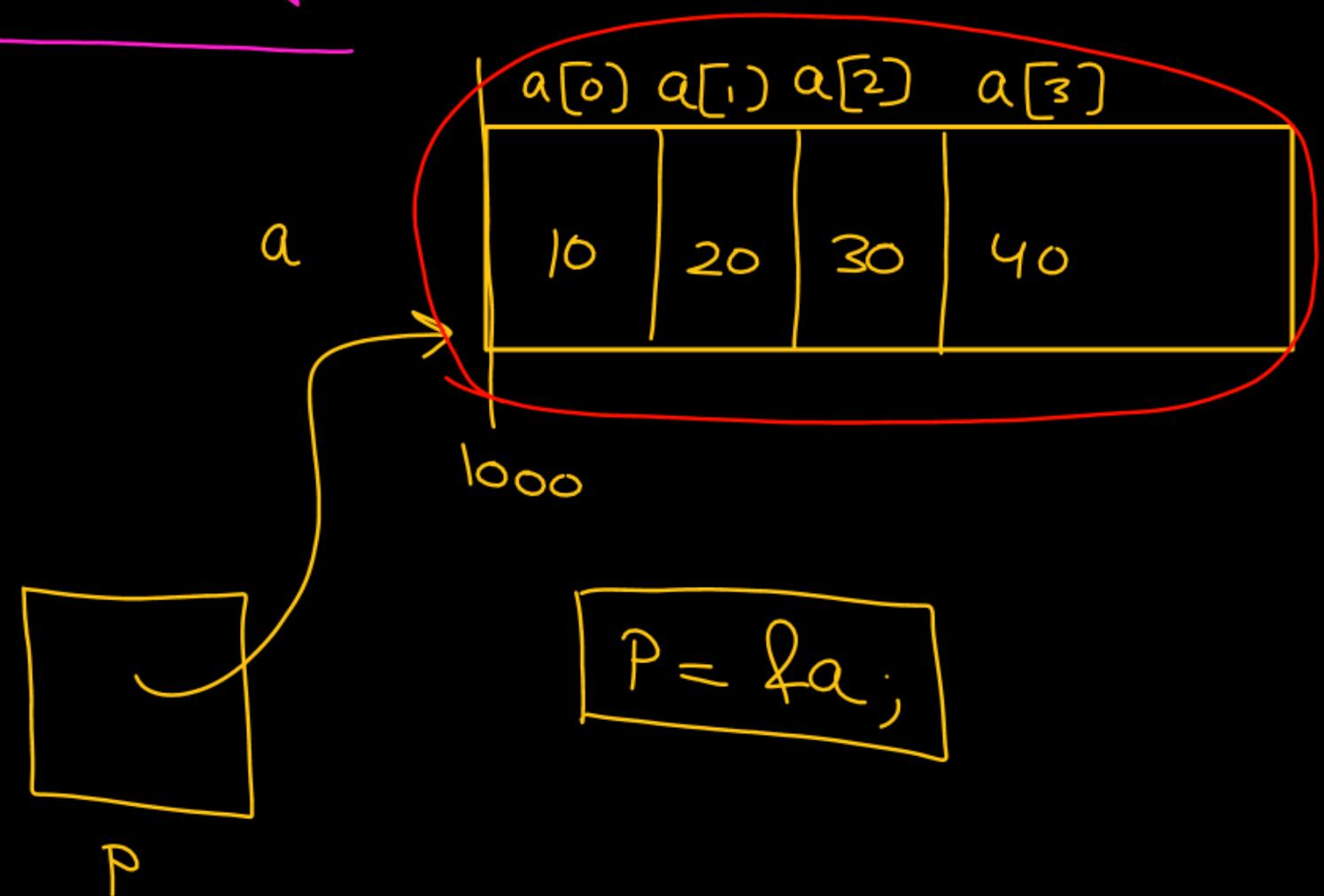
$\text{int } *P[4] = \{a, a+2, a+3, a+1\}$



③ ① ②
int (*P)[4];

P is a pointer to array of 4 integer

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



(3)

③ int (*P)(int, int);

P is a pointer to function

② that takes 2 integer arguments and
① return an integer

③ ~~int Add (int, int);~~

int * f(int *);

f is a function that takes pointer to integer as argument
and returns pointer to integer.

$++$, $*$ \mathcal{R} to L

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

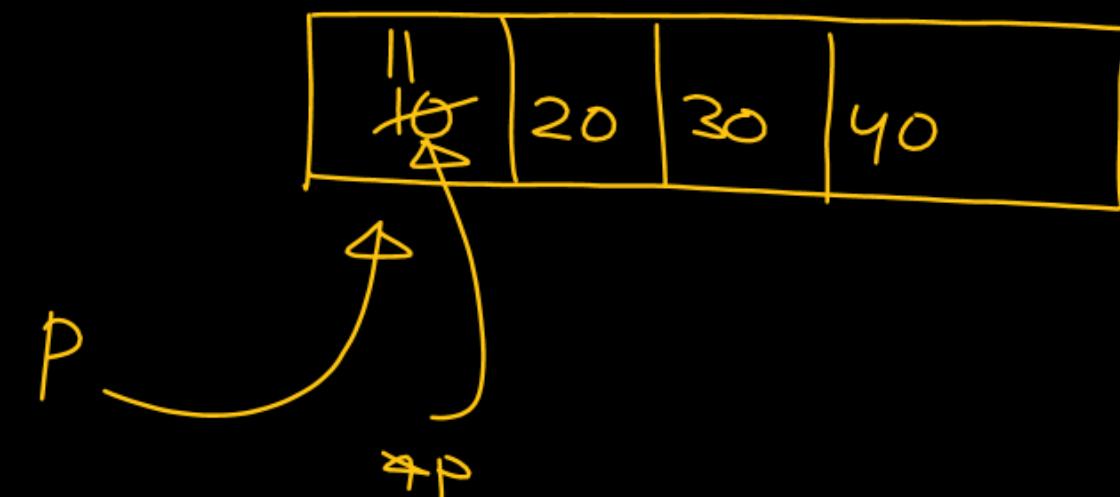
int *p = &a[0];

printf("%d", ++*p); (1)

$++(*p)$

(i) $*p = *p + 1 \Rightarrow *p = 10 + 1 = 11$

(ii) $*p \Rightarrow$ use $\frac{\text{def}}{\text{def}}$



$++P \Rightarrow$ 4ect inc
use | $P++ \Rightarrow$ 4ect use
inc. करे

$++(*P)$

(i) $*P = *P + 1$

(ii) $*P$

void main() {

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

(12);

(12.38);

}

Statement

int a;

a = 10;

expression
statement
with some
value.

$a[0]$	$a[1]$	$a[2]$	$a[3]$
10	20	30	40

$P \sim 100$
 $P = \&a[0]$

20

\checkmark
 $\star P$
 $\times \Rightarrow$ (i) $\star P = \star P + 1$
(ii) use it ($\star P$)

$\star (P + 1)$
(i) $P = P + 1$ (ii) use $\star P$ \times

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

$\text{int } *P;$

$P = \&a[0];$

$\star P;$

$\star \star P;$

$\text{printf}("%d", \underline{\star P});$

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

a[0]	a[1]	a[2]	a[3]
10	20	30	40

int *p[4] = {a, a+1, a+2, a+3};

int **q;

q = &p[0];

++ *q++;

printf("%d", *q) (20)

q = &p[1]

q = ~~&p[1]~~

*q = p[1] = &a[1]

q = ~~&a[1]~~ = a[1]

①

q = &p[0]

*q = p[0]

② p[0] = p[0] + 1

= &a[0] + 1

= &a[1]

q = &p[0] + 1

= &p[1]

&a[1]	&a[0]	&a[1]	&a[2]	&a[3]
p[0]	p[1]	p[2]	p[3]	

&p[1]	&p[0]
q	

Post
inc.

++(*q++)

(i) ++(*q) (ii) q = q + 1

*q = q + 1

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

$a[0]$	$a[1]$	$a[2]$	$a[3]$
10	20	30	40

int $*p[4] = \{a, a+1, a+2, a+3\};$

int $**q;$

$q = \&p[0];$

(31)

Post inc

$++(*q++);$ (i) $++*q$
 $(*q = q + 1;$

$++(*(*q++));$ (ii) $++*q$
printf("%d", $*q$);

$\&a[1]$	$\&a[2]$	$\&a[3]$	$\&a[4]$
$\&a[0]$	$\&a[1]$	$\&a[2]$	$\&a[3]$

$\&p[1]$	$\&p[0]$
----------	----------

q

$$*q = *q + 1$$
$$a[2] = a[2] + 1$$

$$*q = *q + 1$$
$$p[1] = p[1] + 1$$
$$= \&q[1] + 1 = \&q[2]$$

#

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

```
void main( ) {
```

```
    int a[4] = {10, 20, 30, 40};  
    fun(&a[0]);  
    printf("%d", a[0]);  
}
```

a[0]	a[1]	a[2]	a[3]
10	20	30	40

100 104 108 112



$$*P = *P + 1$$

11

```
#_____ int †  
void fun( int *P){  
    ++ *P ;  
}  
  
void main(){  
    int a[4] = {10,20,30,40};  
    fun(a);  
    printf("%d",a[0]); //  
}
```

```

void fun( int (*P)[3] )
{
    ++P; ✓
    ++(*P);
}

```

$$^{**}P = ^{*}P + 1$$

$$a[1][0] = a[0][0] + (^{**}P)$$

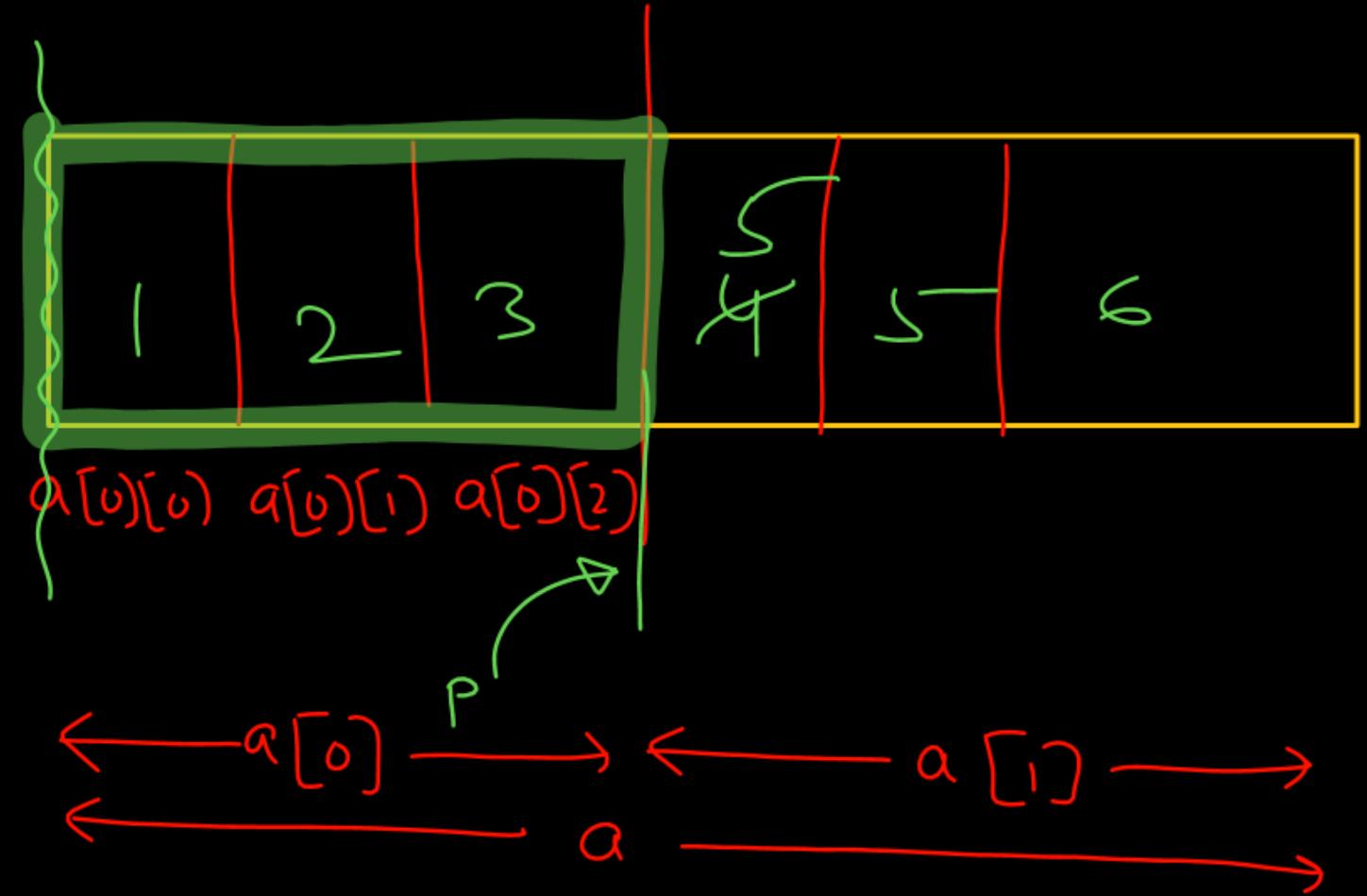
```
void main()
```

```
int a[2][3] = {1,2,3,4,5,6};
```

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

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

```
}
```



$P = \&a[0]$

$^*P = *(\&a[0])$

$^*P = a[0] = \&a[0][0]$

$^{**}P = **\&a[0][0]$

1st chapter

2nd Linked list

array
pointer
structure

CDSA \Rightarrow

\Rightarrow DS



Tommorow \Rightarrow off

++q

inc/dec \Rightarrow result constant

$\text{++}(r++) \times$

