CS & IT ENGINEERING

C Programming

Arrays and Pointers

Lec - 07

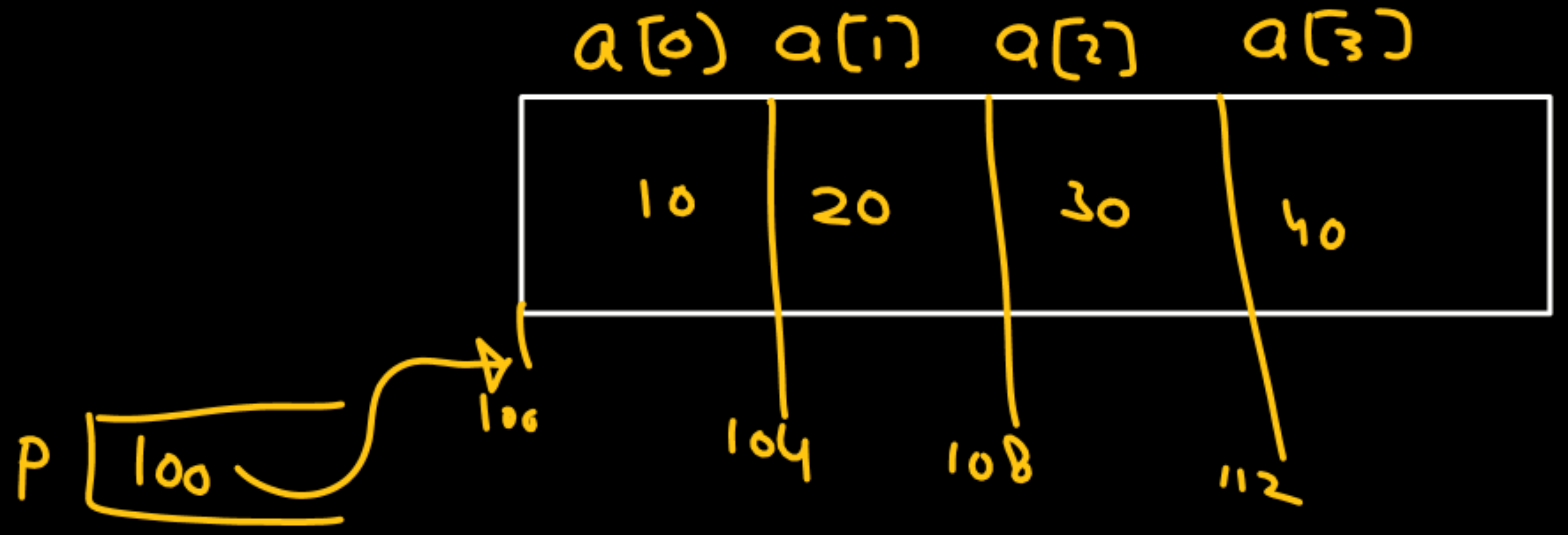
By- Pankaj Sharma Sir

TOPICS TO BE COVERED

Arrays and Pointers (Part- 07)

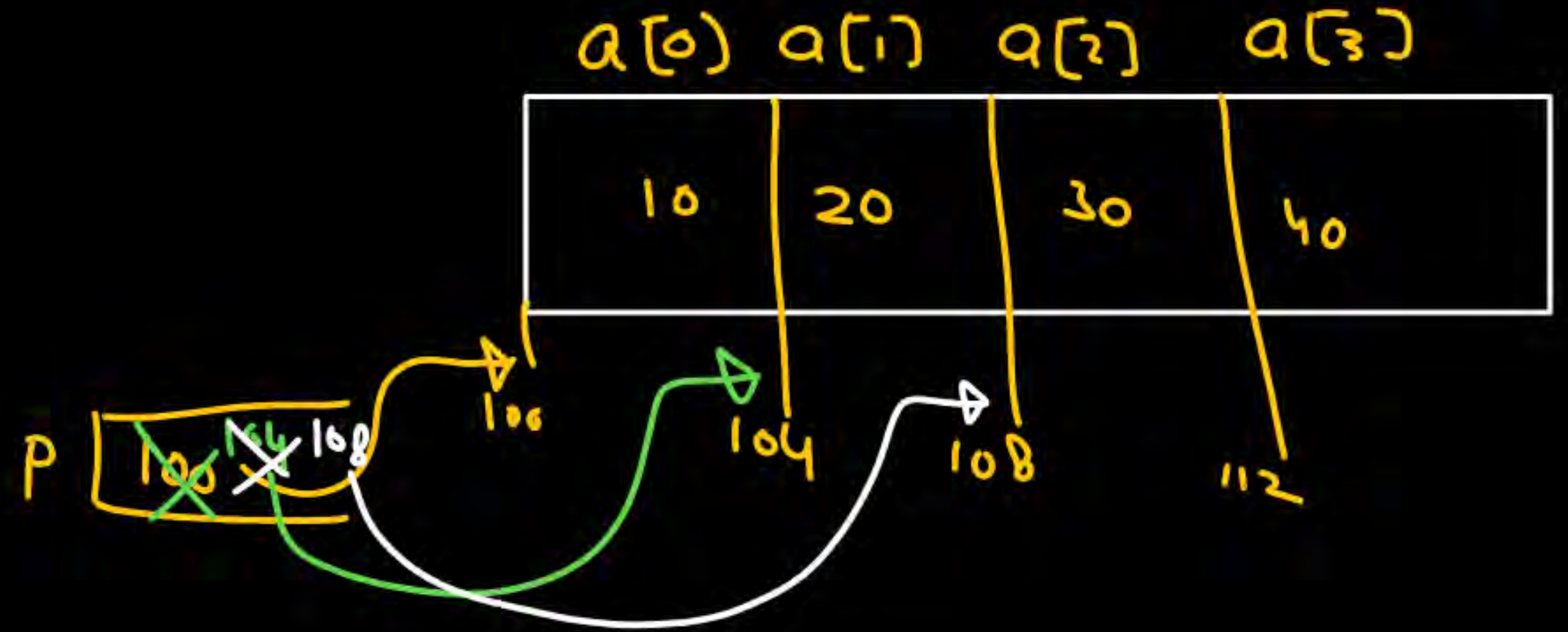
```
void main(){

    int a[4] = {10,20,30,40};
    fun(a);
    printf("%d",a[1]);
            }
```

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

}
```



```
void main(){

    int a[4] = {10,20,30,40};
    fun(a);
    printf("%d", a[1]);

        }
```

```
void fun( int *p)
  {

      ++p;

   *(p++);   ---> (i) *p
                  (ii) p=p+1

  }
```



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

```
void fun(int *p) ✓
   {
    =
   }
```

```
void fun(int p[ ]) ✓
   {
    =        int *p
   }
```

```
void main(){

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

    sum = fun(a,4);

    pf("%d", sum);
      }
```

```
int fun(int *p , int n)
  {
   int  i,sum=0;

   for (i=0; i<n; i++)
         sum = sum+p[i];

   return sum;

   }
```

Complex declaration

1	( )	paranthesis	function	1	L to R
2	[ ]	square bracket	Array		
3	Identifier	var. name, fun. name		2	R to L
4	*	Pointer			
5	data type	int, char, float, ...		3	

int *p $\longrightarrow$ (i) int is a star p ✗

(ii) star is a int p ✗

(iii) $\textcircled{p}$ is a pointer to int. ✓

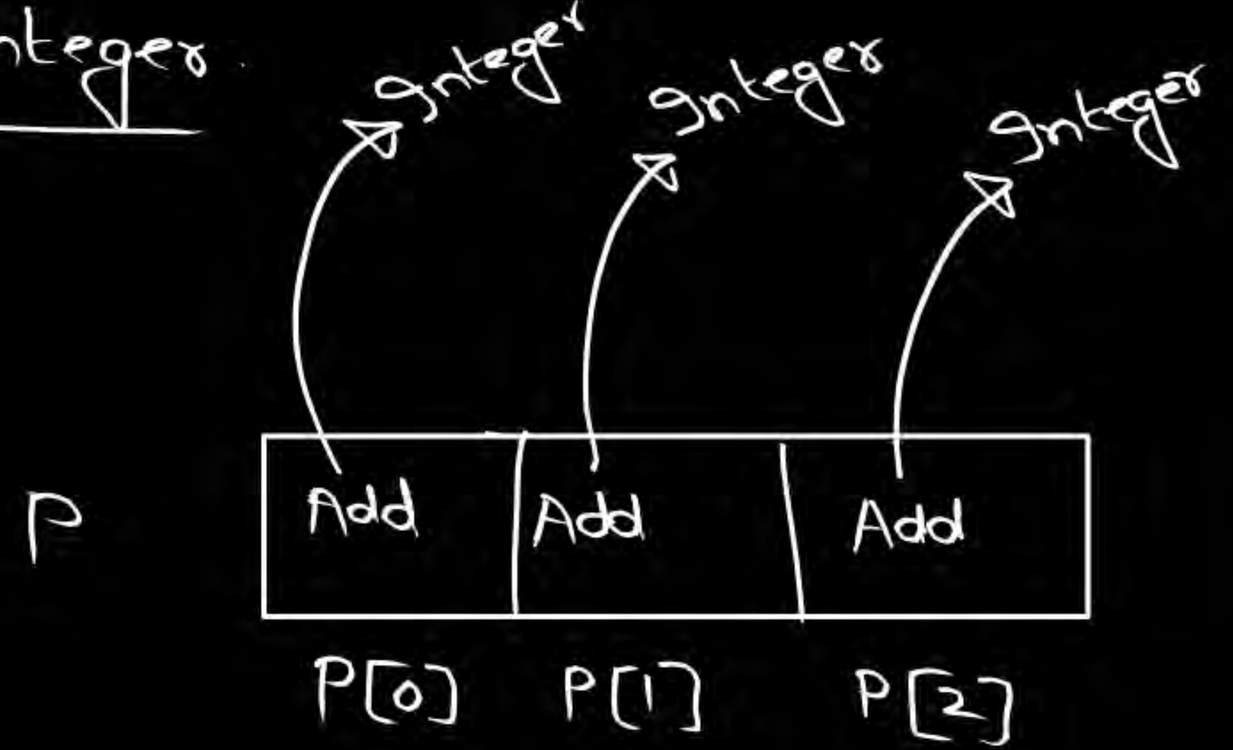
$\downarrow$

Identifier

(i) $\overset{③}{\text{int}}\ \overset{②}{*}\ (\overset{①}{P[3]})$

P is an array of 3 Pointer to integer

P is an array of 3 pointer to integer.

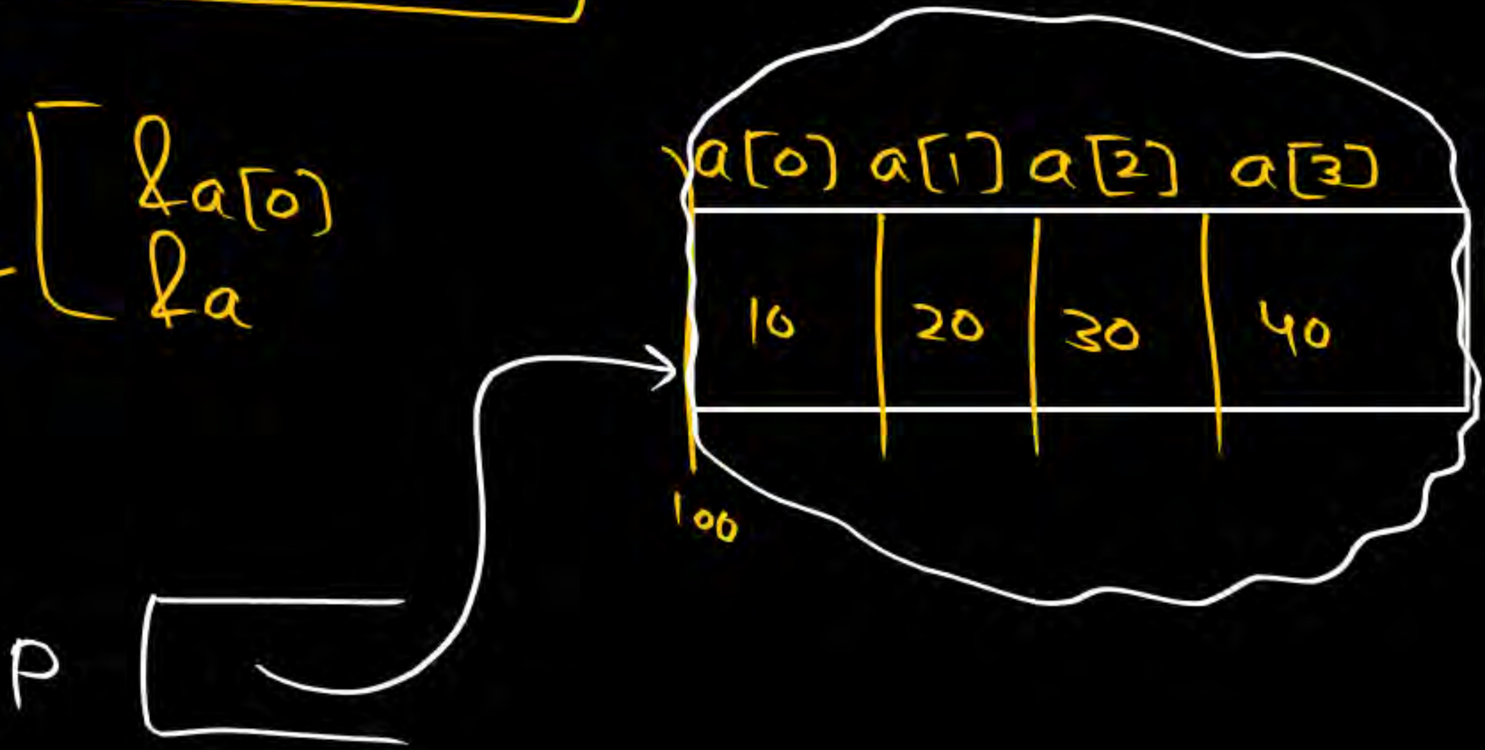


(ii) int $\overset{1}{(*p)}\overset{2}{[4]}$; with 3 over int

p is a pointer to $\boxed{\text{array of 4 integer}}$

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

numerical [&a[0]
logically diff. ← [&a



(iii) $\overset{③}{\text{int}}\ (\overset{①}{*P})\overset{②}{(\text{int}, \text{int})};$

P is a pointer to a function that takes 2 integer argument and return an integer value.

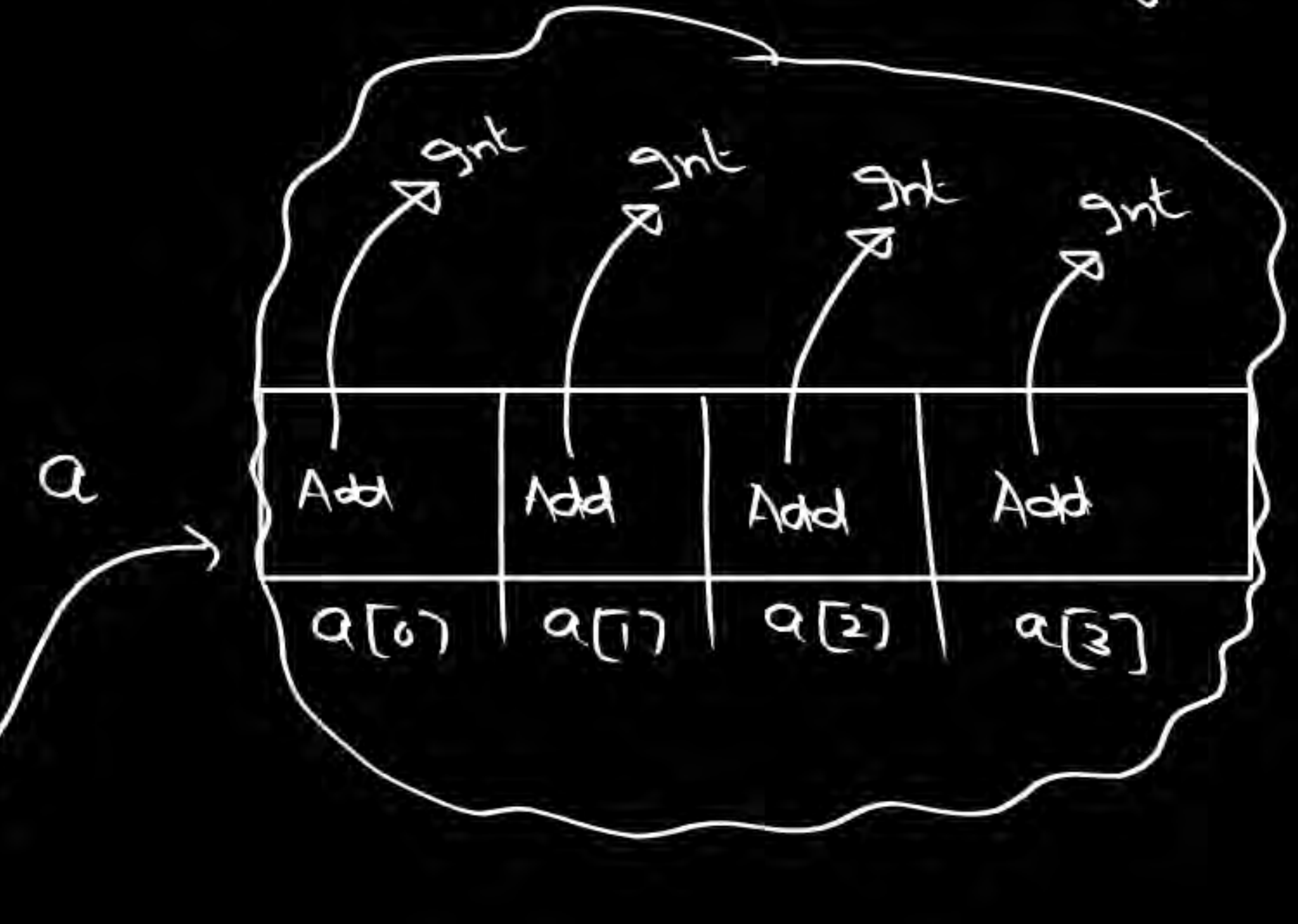
$\underset{\text{return type}}{\underbrace{\text{int}}}\ \text{Add}(\underset{\text{arg. list}}{\underbrace{\text{int}, \text{int}}});$

(IV) int * (*p)[4];

③ ① ②

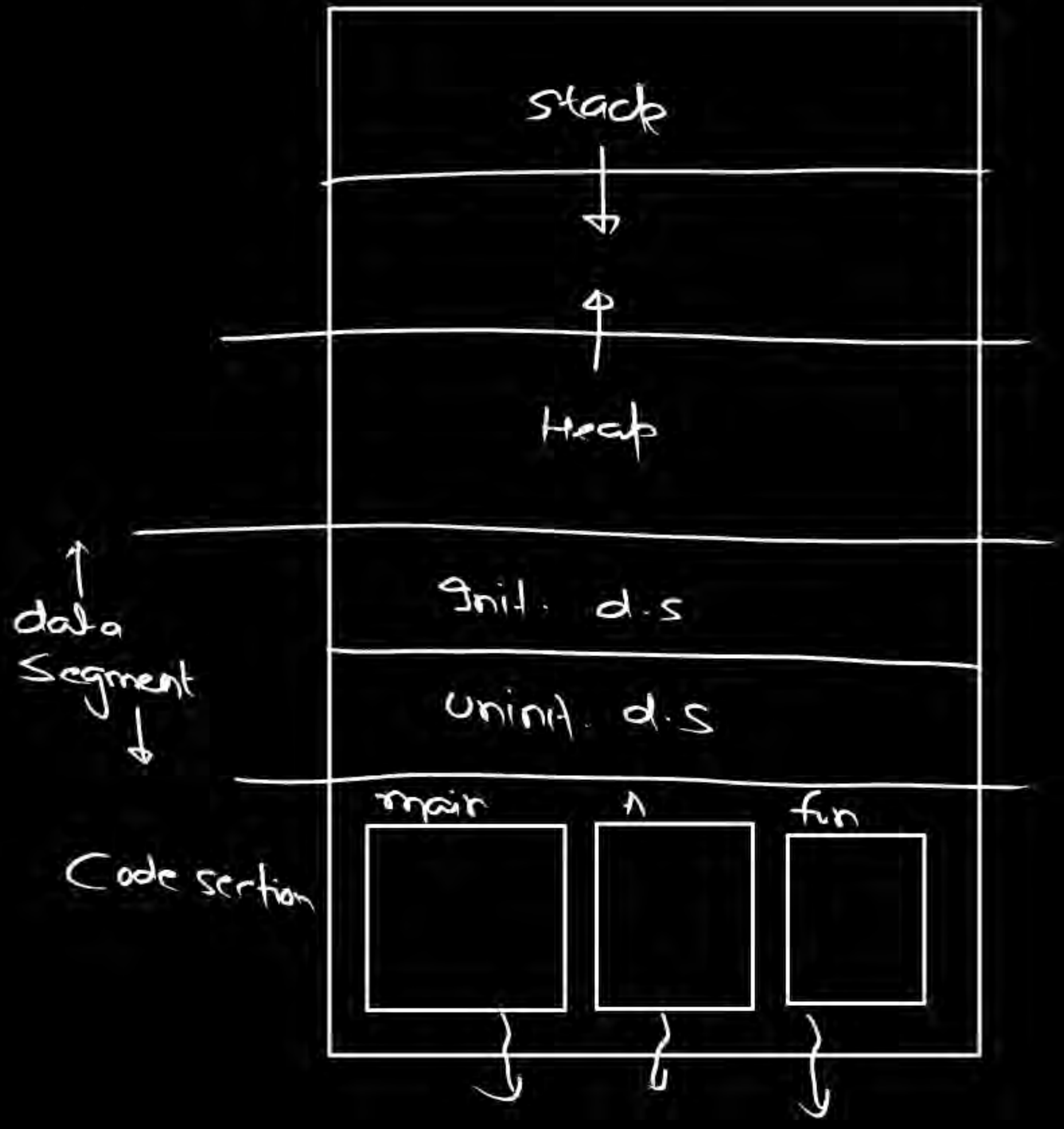
p is a pointer to an array of 4 pointer to integer

p = &a



Pointer to function / function pointer

```
void fun( ){
        =
        -
        }
void  A( ) {
        -
        =
        -
        }
void main( ){
        =
        -
        }
```



$$\begin{tikzcd}
 & \text{arg list} \\
\text{function} \arrow[ur] \arrow[dr] & \\
 & \text{return type}
\end{tikzcd}$$

$$\boxed{\texttt{int Add(int , int) ;}}$$

$$\overset{3}{\texttt{int}}\ (\overset{1}{\texttt{*p}})\ (\overset{2}{\texttt{int, int}});$$

p is a pointer to function that takes 2 integer arg and return integer.

```
int Add(int a, int b)
      {
        return a+b ;
      }
 void main(){
        int (*P)(int , int);
        P =&Add ;
printf("%d",(*P)(2,3));
          }
```

```
int Add(int a, int b)
{
    return a+b ;
}
void main(){
    int (*P)(int , int);
    P =&Add ;
Call ⟹  (*P)(2,3) ;
}
```

①
```
main( ){
    P = Add ;
    (*P)(2,3) ; ✓
}
```

②
```
main(){
P = Add ; ✓
(P)(2,3) ; ✓
}
```

③
```
main(){
    P = &Add ; ✓
    (*P)(2,3) ; ✓
}
```

④
```
main(){
    P = &Add ;
    (P)(2,3) ;
}
```

```
int (*P)(int, int);

int Add(int, int);
int diff(int, int);
int mul(int, int);
int div(int, int);

    P = Add;
    ==
    P = diff;
      ==
    P = mul;
      ==
    P = div;
```

Q.

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

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

int **q;

q = &P[0];

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

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

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

q | ~~&P[0]~~ &P[1]

$*++*(++q) \longrightarrow$ (i) $q = q+1$

(ii) $*++*q$

Q.

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

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

int **q;

q = &p[0];

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

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

&a[2]	~~&a[0]~~ &a[1]	&a[1]	&a[3]
P[0]	P[1]	P[2]	P[3]

q	~~&P[0]~~ &P[1]

$*++*(++q) \longrightarrow$ (i) $q = q+1$

(ii) $*++*q$

$*(++(*q))$

↓

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

(ii) $**q$

$P[1] = P[1]+1$

$= \&a[0]+1$

$= \&a[1]$

Q.

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

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

int **q;

q = &p[0];

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

$\boxed{20}$

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

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

q | ~~&p[0]~~ &p[1]

$*++*(++q) \rightarrow$ (i) $q = q+1$
(ii) $*++*q$

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

$p[1] = p[1]+1$
$= \&a[0]+1$
$= \&a[1]$

pf("%d", **q)

$*\cancel{*\&}p[1] \Rightarrow *p[1] \Rightarrow \cancel{*\&}a[1] = a[1]$

Q

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

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

int **q ;

q = &P[0] ;

*++ *++q ;        done ✓

--q ;

pf("%d", *--*++q);
```

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

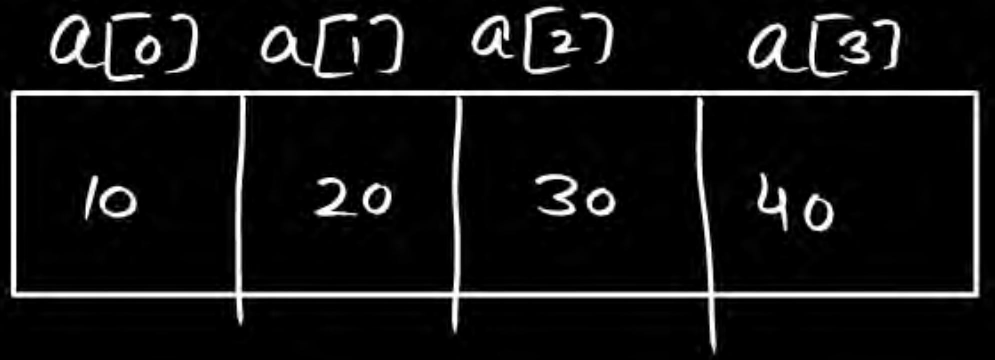
P[0]	P[1]	P[2]	P[3]
&a[2]	&a[1] ~~&a[0]~~	&a[1]	&a[3]

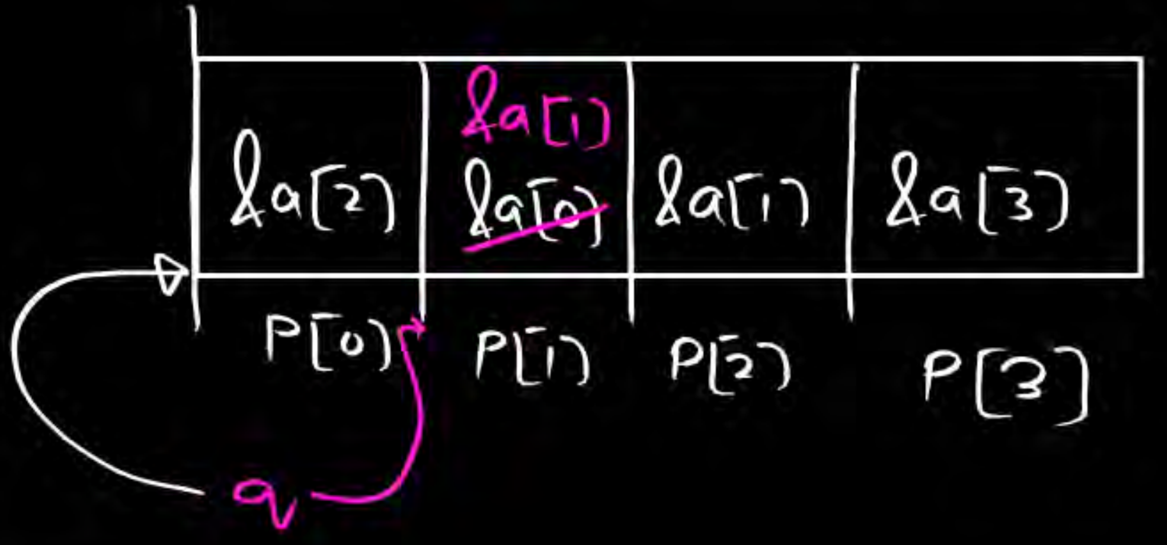
q	
q	~~&P[0]~~ &P[1]

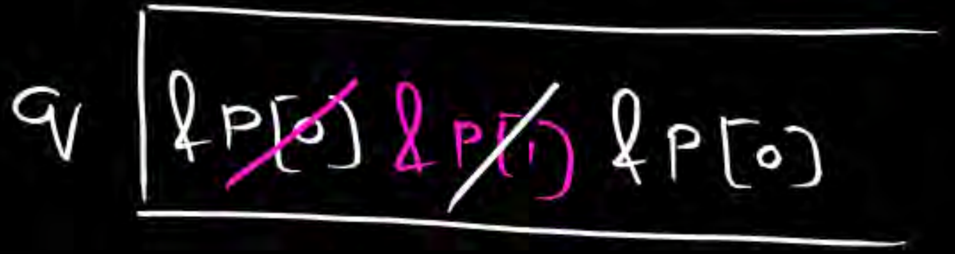
```
*++*(++q)
(i) q = q+1
(ii) *++*q
        ↓
*++(*q)
(i) ++(*q) ⇒ *q = *q+1
(ii) *q ⇒ useless
```

Q

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

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

int **q;

q = &P[0];

*++*++q;        done ✓

--q; ✓

pf("%d", *--*++q);
```

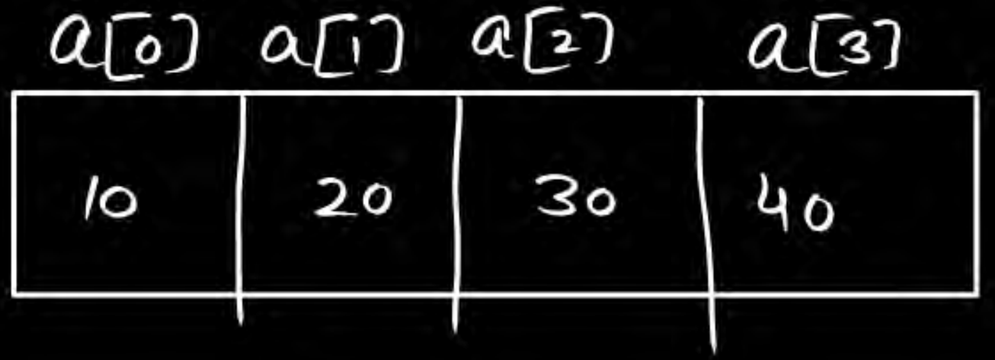


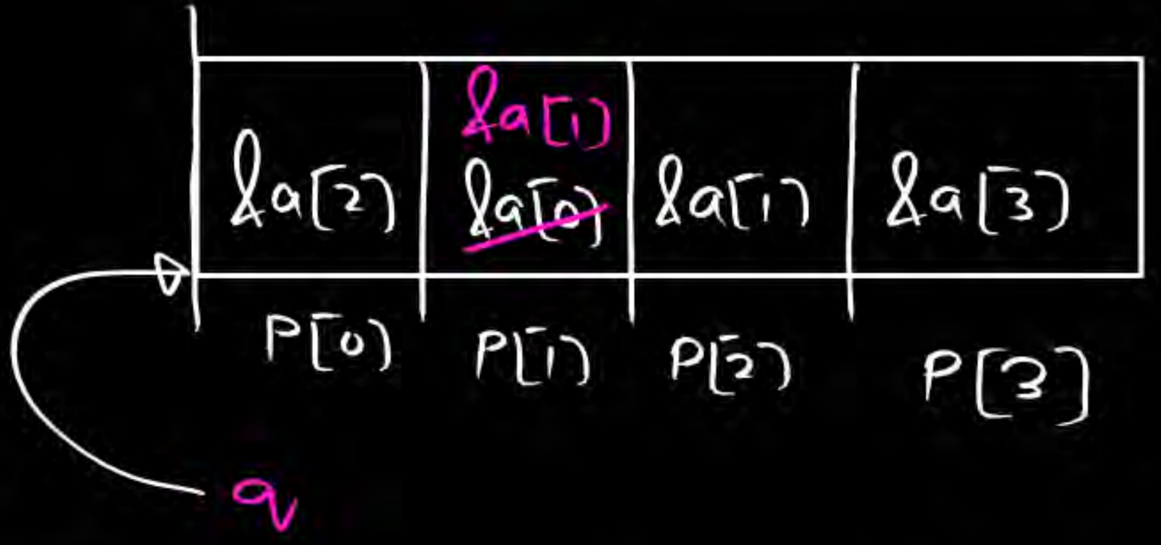


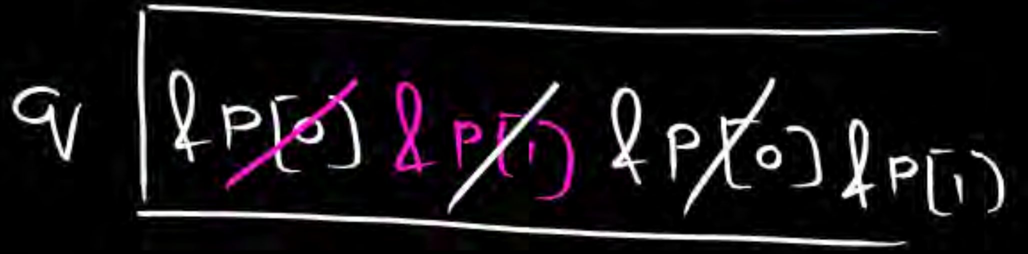


Q.

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

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

int **q ;

q = &p[0];

*++*++q ;    done ✓

--q ; ✓

pf("%d", *--*++q);
```







`*--*(++q)`

(i) q = q+1

(ii) `*--*q`

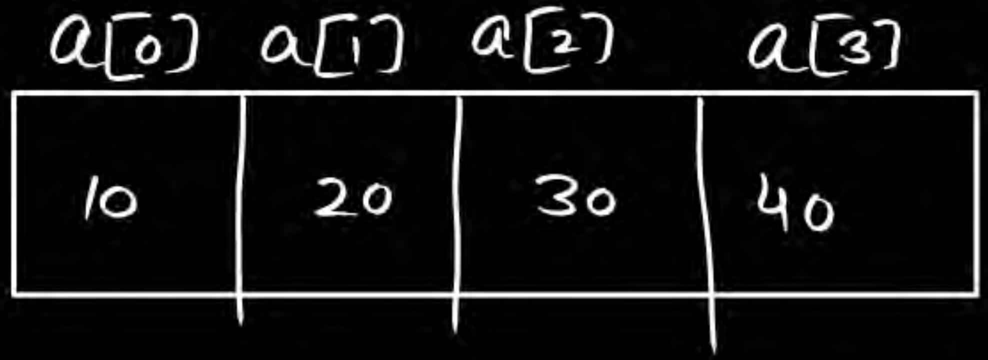
`*(--(*q))`

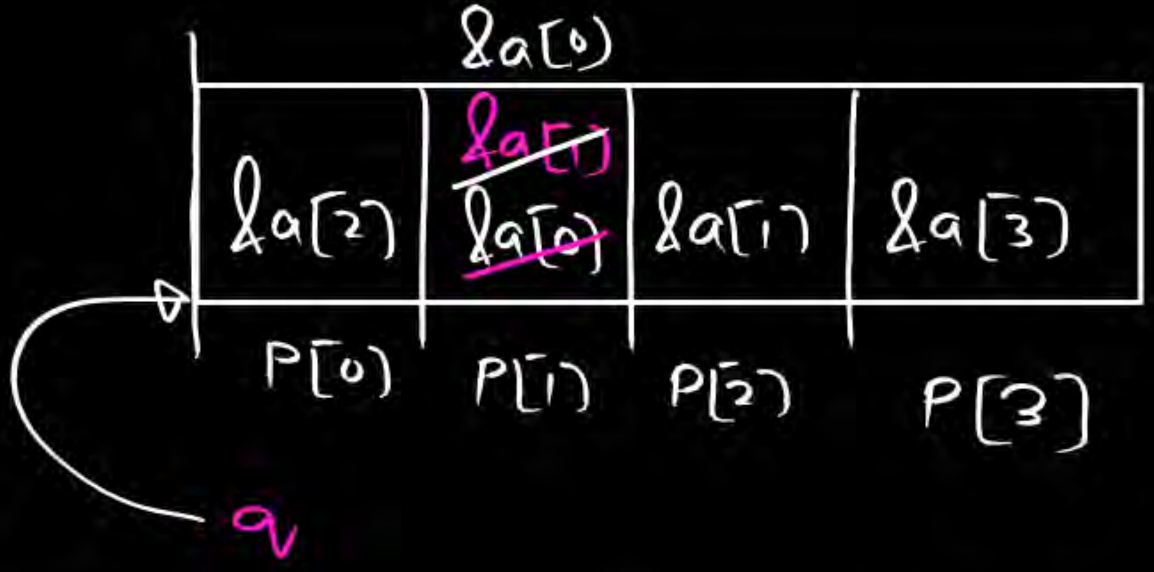
(i) `*q = *q-1`

(ii) `**q`

Q/

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

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

int **q;

q = &P[0];

*++*++q ;        done ✓

--q ; ✓

pf("%d", *--*++q);
   6

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





q | ~~&P[0]~~ ~~&P[1]~~ ~~&P[0]~~ &P[1]

`*--*(++q)`

(i) q = q+1

(ii) `*--*q`

`*(--(*q))`

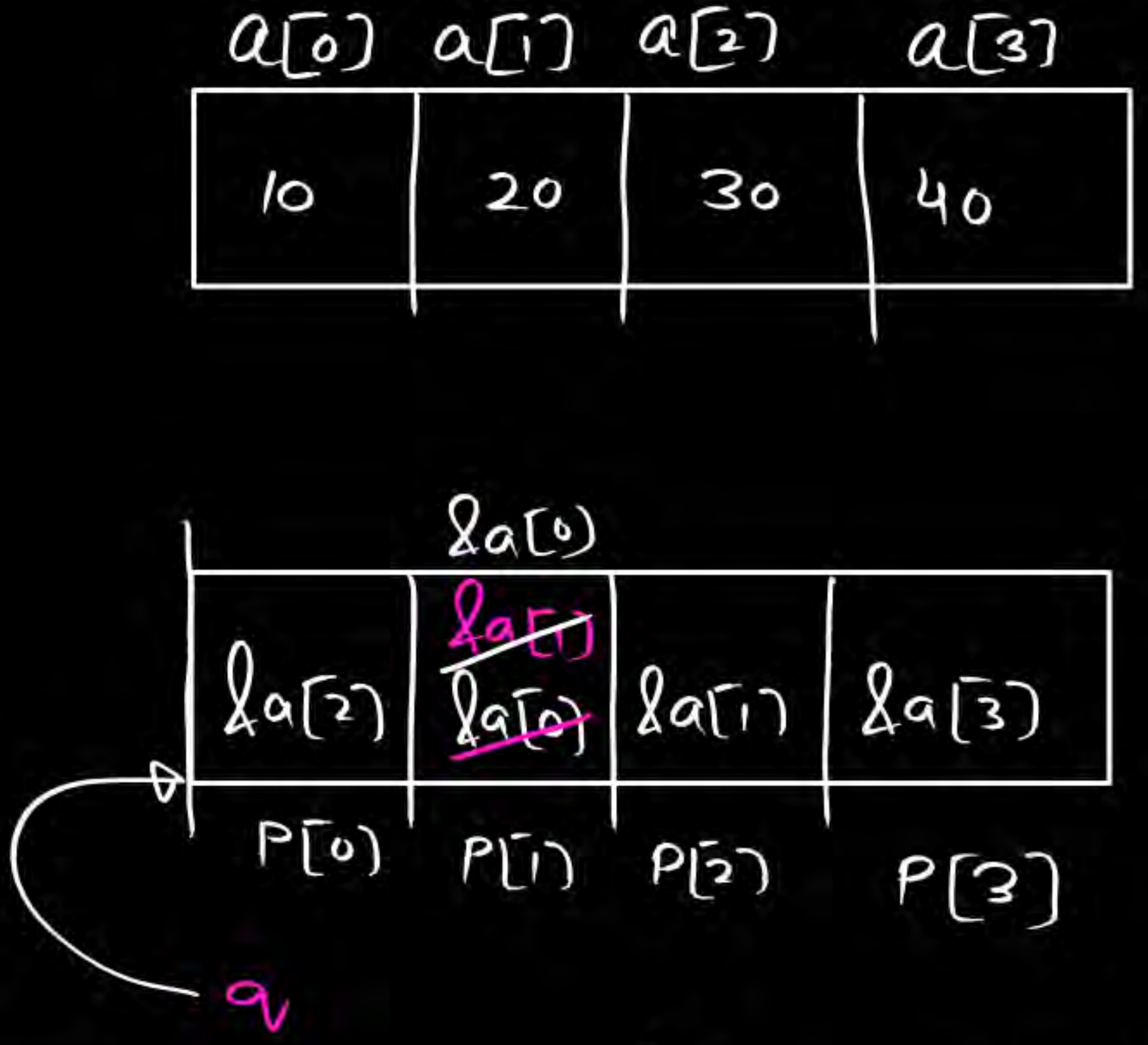
(i) `*q = *q-1`

(ii) `**q`

P[1] = P[1]-1
= &a[1]-1
= &a[0]

Q.

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

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

int **q;

q = &p[0];

*++*++q;        done ✓

--q; ✓

pf("%d", *--*++q);

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



q | ~~&p[0]~~ ~~&p[1]~~ ~~&p[0]~~ &p[1]

$$**q = *\cancel{*}\cancel{\&}p[1] = *p[1] = \cancel{*}\cancel{\&}a[0] = a[0]$$

$\boxed{10}$

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

Q

```
void fun(int *);

void main(){

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

    fun(a);

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

void fun(int *p){

        ++p;  → p  point to &a[1]
        *p++;
        ++*p;
        }
```

p | ~~&a[0]~~ &a[1]

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

Q/

```
void fun(int *);

void main(){

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

    fun(a);

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

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

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

$\rightarrow$ *(p++)

(i) *p ⇒ useless

(ii) p = p+1

a[0]	a[1]	a[2]	a[3]
10	20	31 ~~30~~	40

Q

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

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

$\rightarrow$ ++(*p)

(i) *p = *p + 1

(ii) *p $\Rightarrow$ useless

Q

```
void main(){
int a[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
   fun(a);
pf("%d %d %d", a[1][1], a[1][2], a[2][0]);
      }

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

         ++p;
      (*p)[1] = (*p)[1]+1;

               }
```

&a[0]

address of (array of 3 integer)

↓

Pointer to (array of 3 int.)

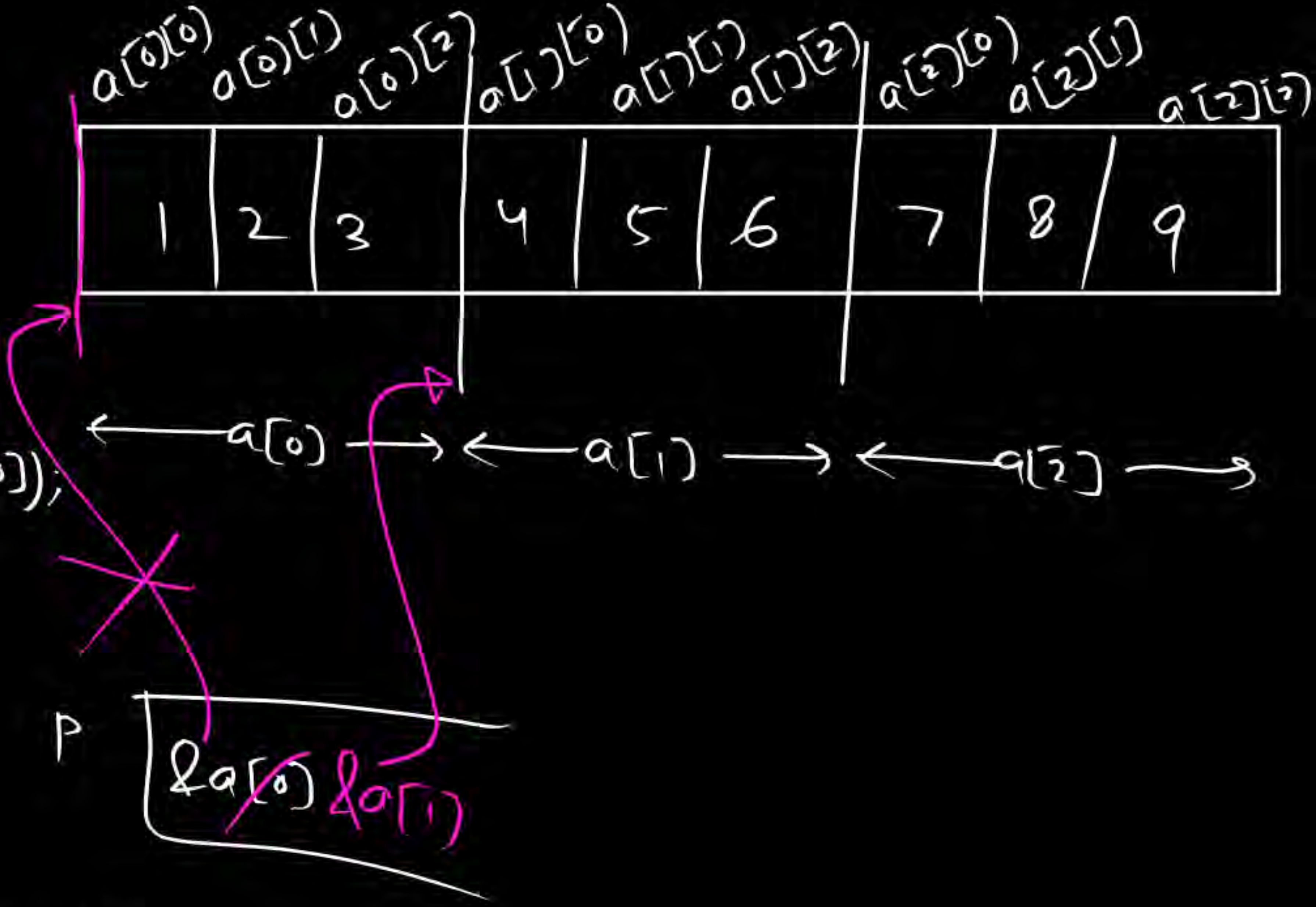
↓

int (*p)[3]

Q

```
void main(){
 int a[3][3] = {{1,2,3},{4,5,6},{7,8,9}};

   fun(a);

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

      }

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

        ++p;
    (*p)[1] = (*p)[1]+1 ;

        }
```

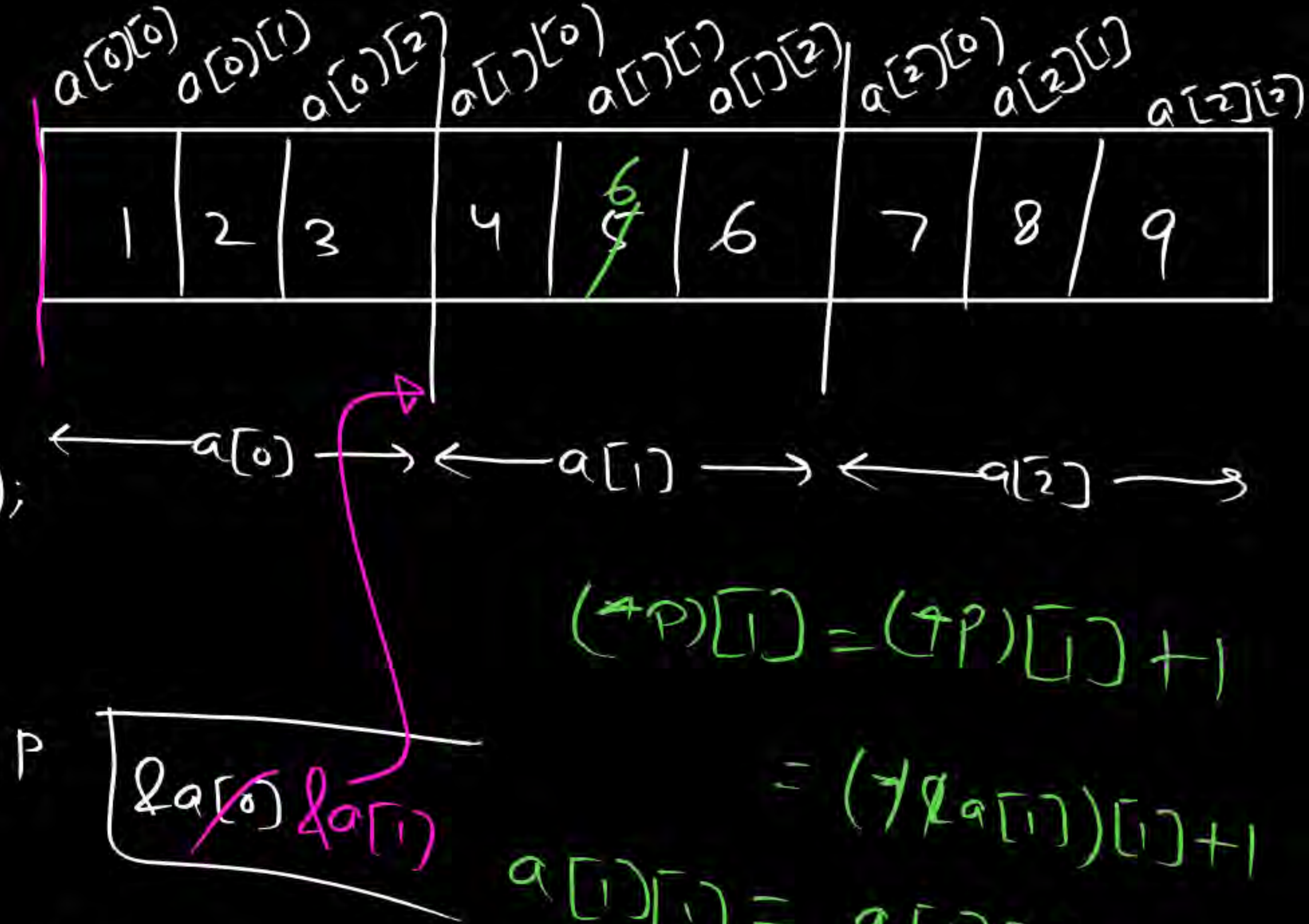
$$P = p+1$$
$$= \&a[0]+1$$
$$= \&a[1]$$



Q

```
void main(){
    int a[3][3] = {{1,2,3},{4,5,6},{7,8,9}};

      fun(a);
                        6       6       7
  pf("%d %d %d", a[1][1], a[1][2], a[2][0]);

        }

void fun( int (*P)[3]) {

        ++P;
    (*P)[1] = (*P)[1] + 1 ;

        }
```

$$P = P+1$$
$$= \&a[0]+1$$
$$= \&a[1]$$



$$(*P)[1] = (*P)[1]+1$$
$$= (*\&a[1])[1]+1$$
$$a[1][1] = a[1][1]+1$$

THANK YOU GW SOLDIERS !