CS & IT ENGINEERING

C Programming

Arrays and Pointers

Lec - 05

By- Pankaj Sharma Sir

TOPICS TO BE COVERED

Arrays and Pointers (Part- 05)

```
int x = 10;
```

```
int *p;
p = &x;
```

```
int *p = &x;
```

```
int x;
x=10;
```

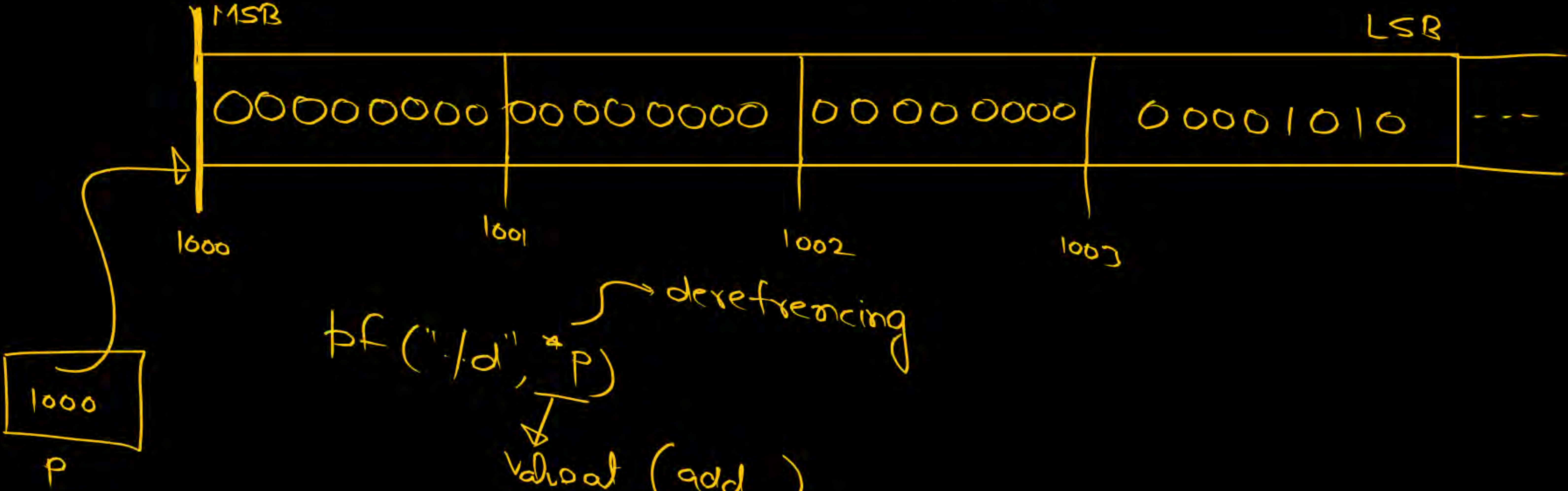
initialization

```
int x=10;
```

```
int *p;
p=&x;
```

⇒

```
int *p = &x;
```

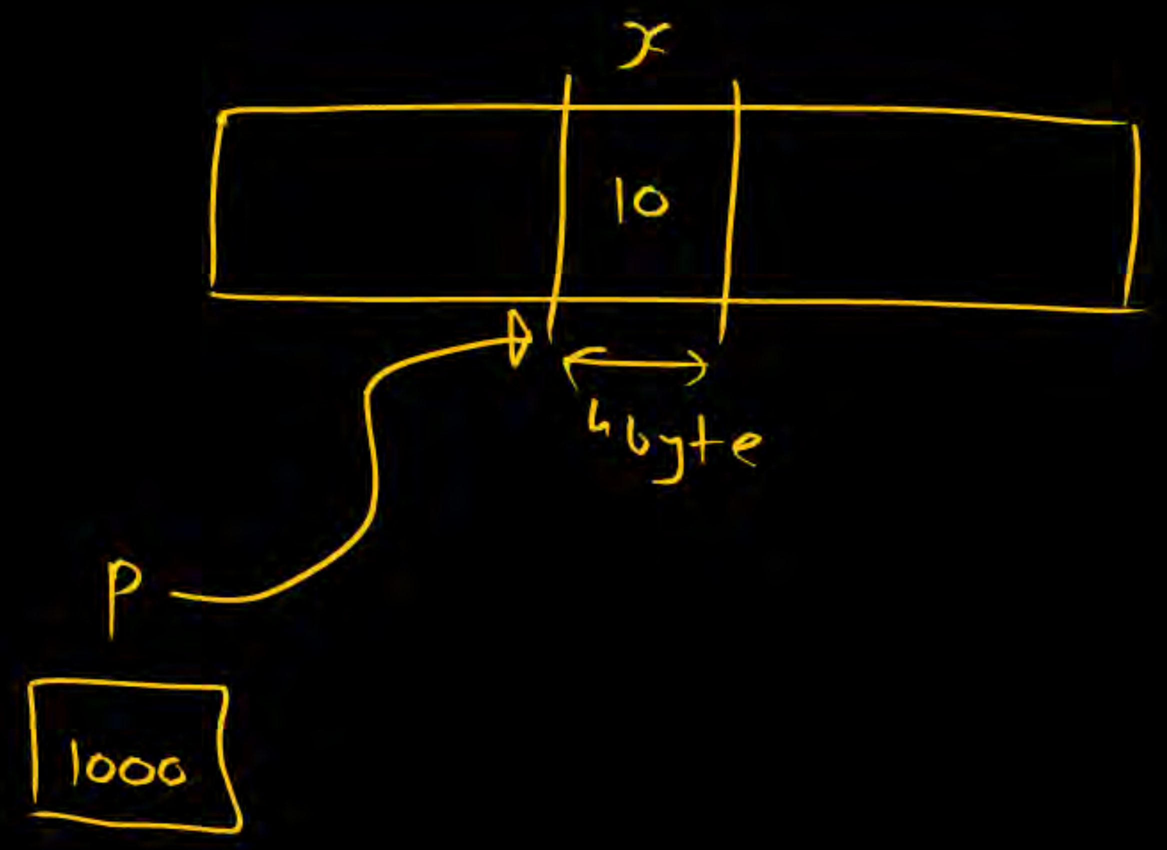


```
pf("%d", *p)
```

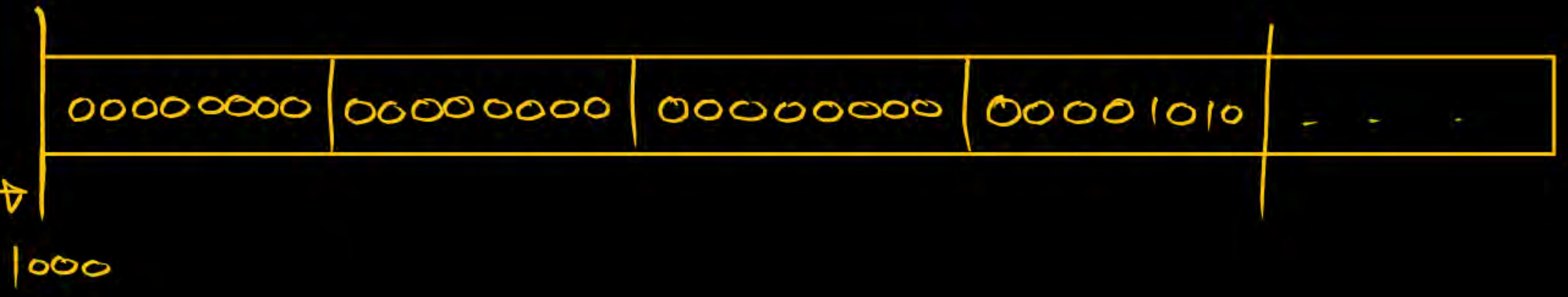
→ derefrencing

Valoat (add)

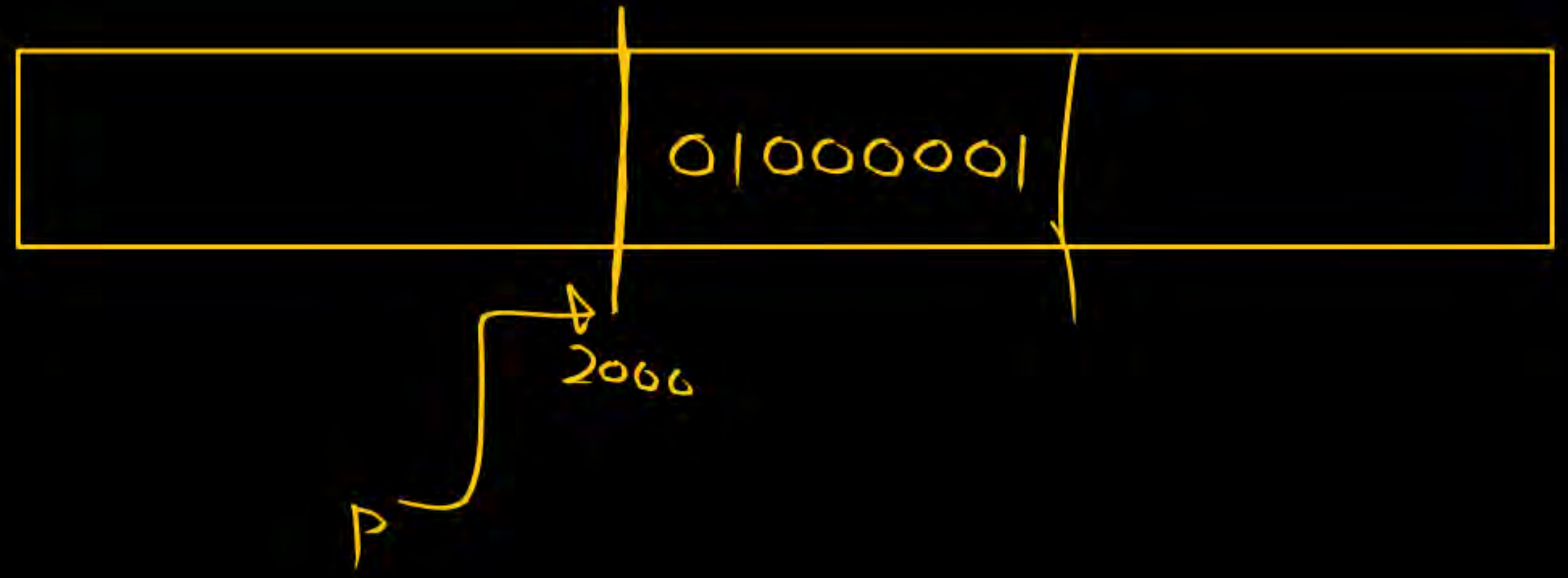
```
int x=10;
int*p ;
```



```
1
int *p;
p = &x;
pf("%d", *p);

4 byte
```



```
2
char *p;
char x = 'A';
p = &x;
pf("%c", *p);
```



int - 4byte

char - 1byte

float → 8byte

int *p; → p → 4 byte

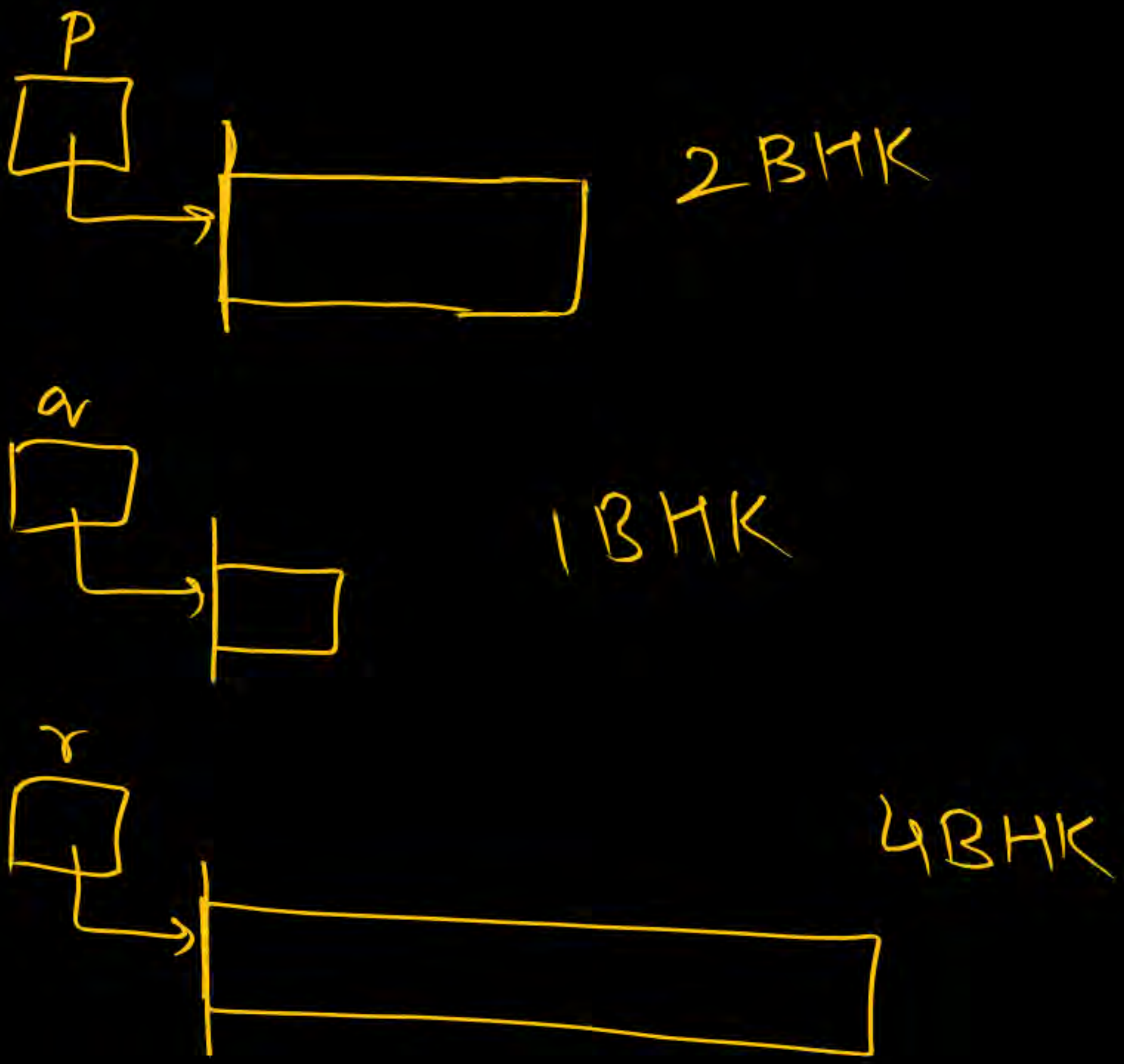
char *q; q → 1 byte

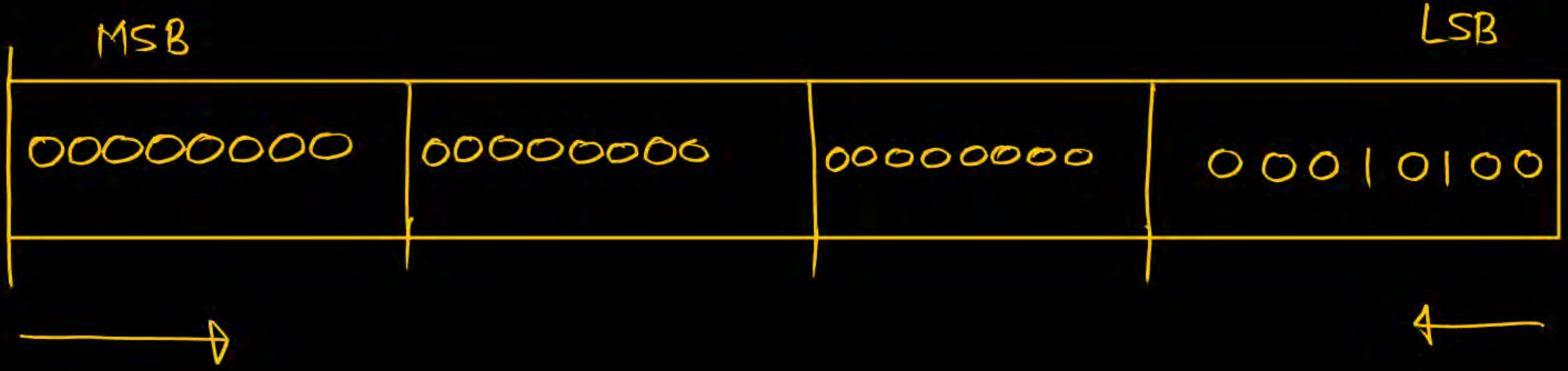
float *r; r → 8 byte

p =

q =

r =





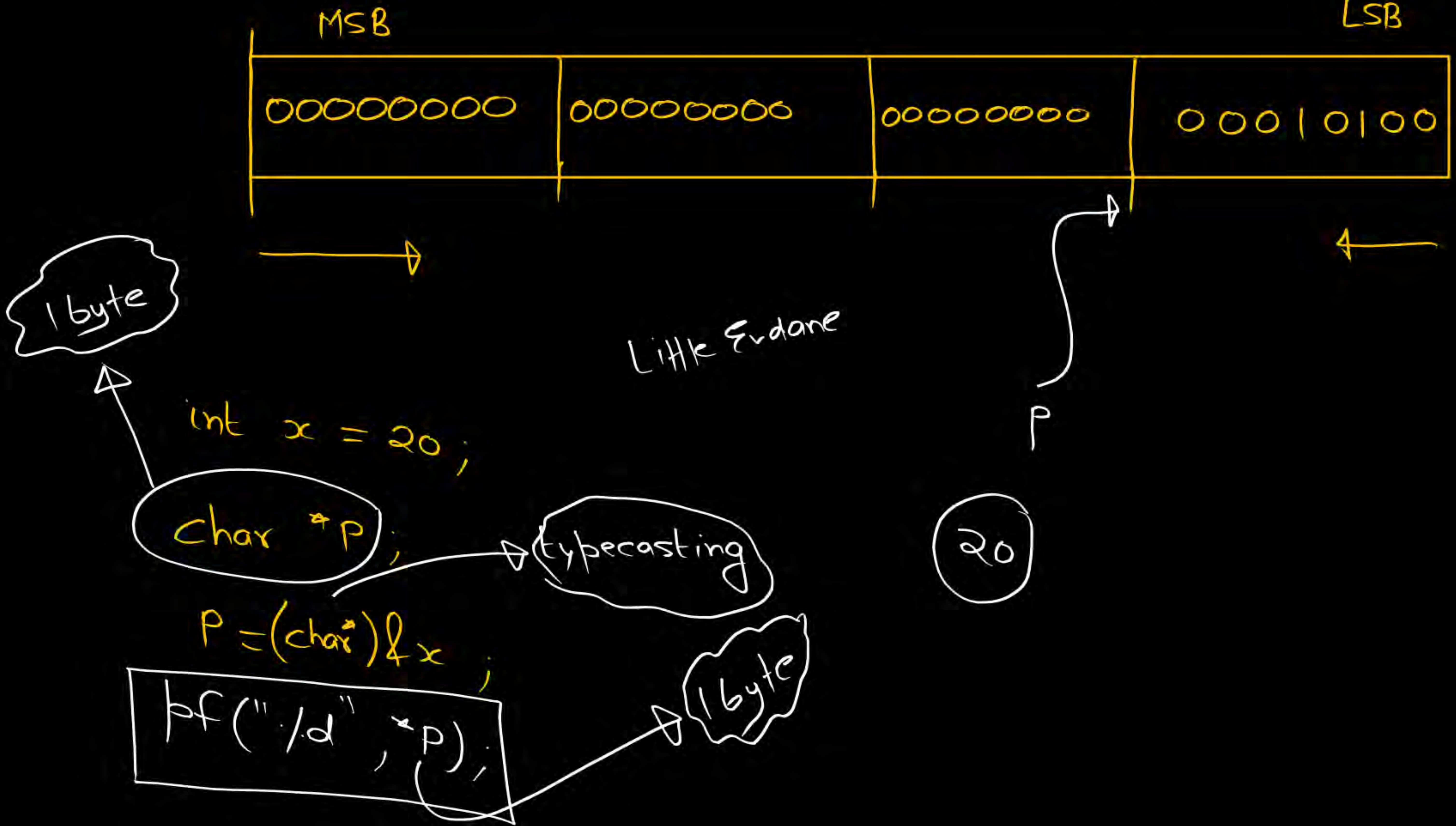
000000000	000000000	00000000	00010100

```
int x = 20;
int *p = &x;
```

Little Endane

Big Endane

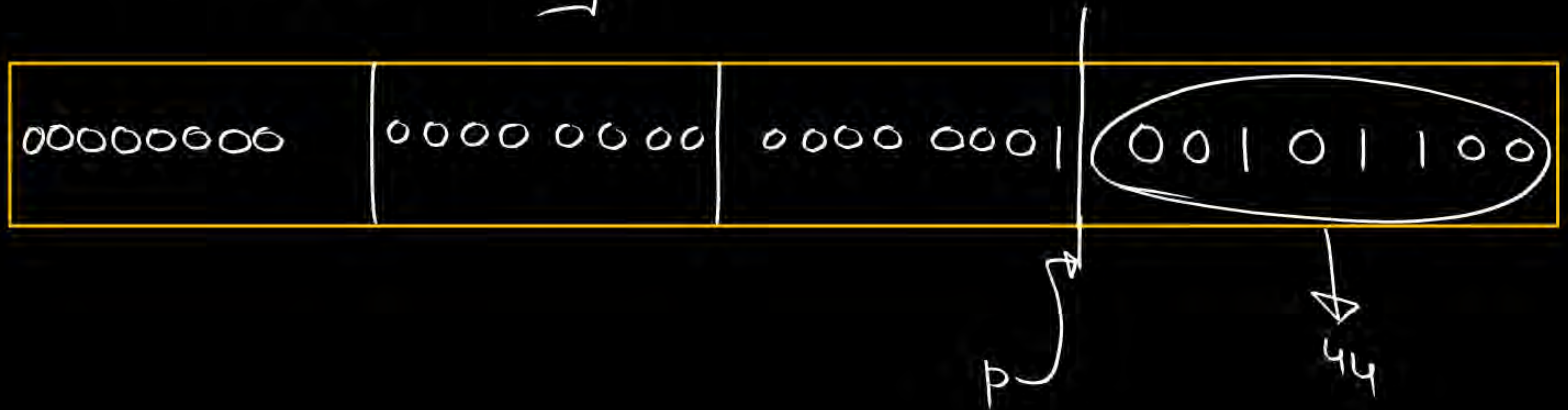
MSB is stored first



```
int x = 20;
char *p;
p = (char*)&x;
pf("%d", *p);
```

```
int x = 300 ;
char *p ;
p = (char *)&x ;
pf("%d", *p);
```

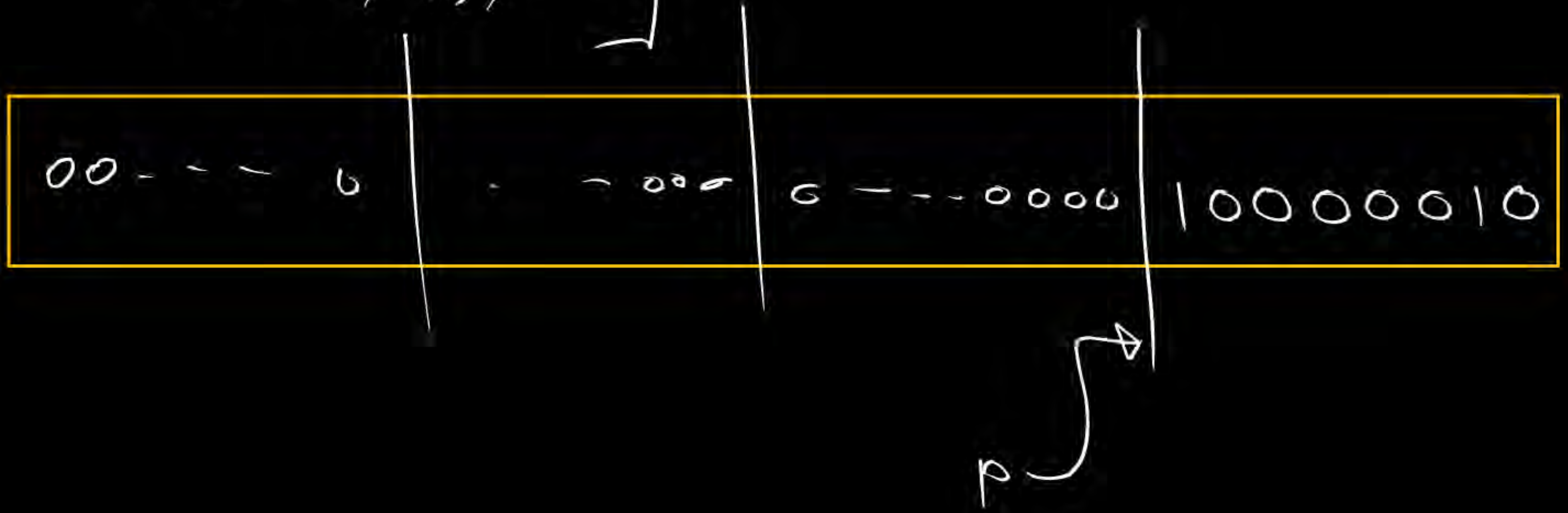
256
32
8
4



00000000	0000 0000	0000 0001	0010 1100

```
int x = 130; ✓
char *p ;
p = (char *)&x ;
pf("%d", *p);
```

128
2



$$\begin{array}{cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array}$$

$$\boxed{10000010}$$

—ve

2's comp form

$$\Rightarrow -2^6 - 2^5 - 2^4 - 2^3 - 2^2 - 2^0 - 1$$

$$= -64 - 32 - 16 - 8 - 4 - 1 - 1$$

$$= -96 - 30$$

$$= \boxed{-126}$$

```
int x = 400;
char *p;
p= (char*)&x;
pf("%d", *p);
```

$$\left[\begin{array}{r} 256 \\ 128 \\ 16 \end{array}\right. \qquad 384$$

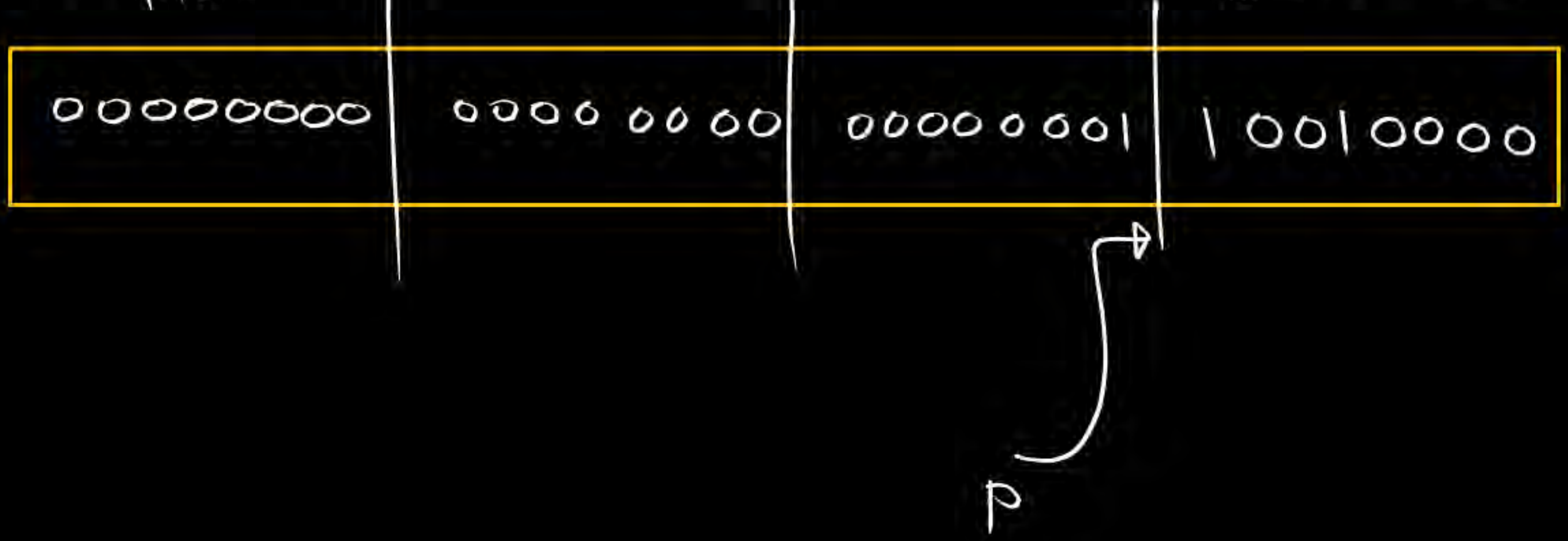
$$\begin{array}{cccccccc} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array}$$

$$= -2^6 - 2^5 - 2^3 - 2^2 - 2^1 - 2^0 - 1$$

$$= -64 - 32 - 8 - 4 - 2 - 1 - 1$$

$$= -96 - 16 \Rightarrow \boxed{-112}$$

MSB LSB



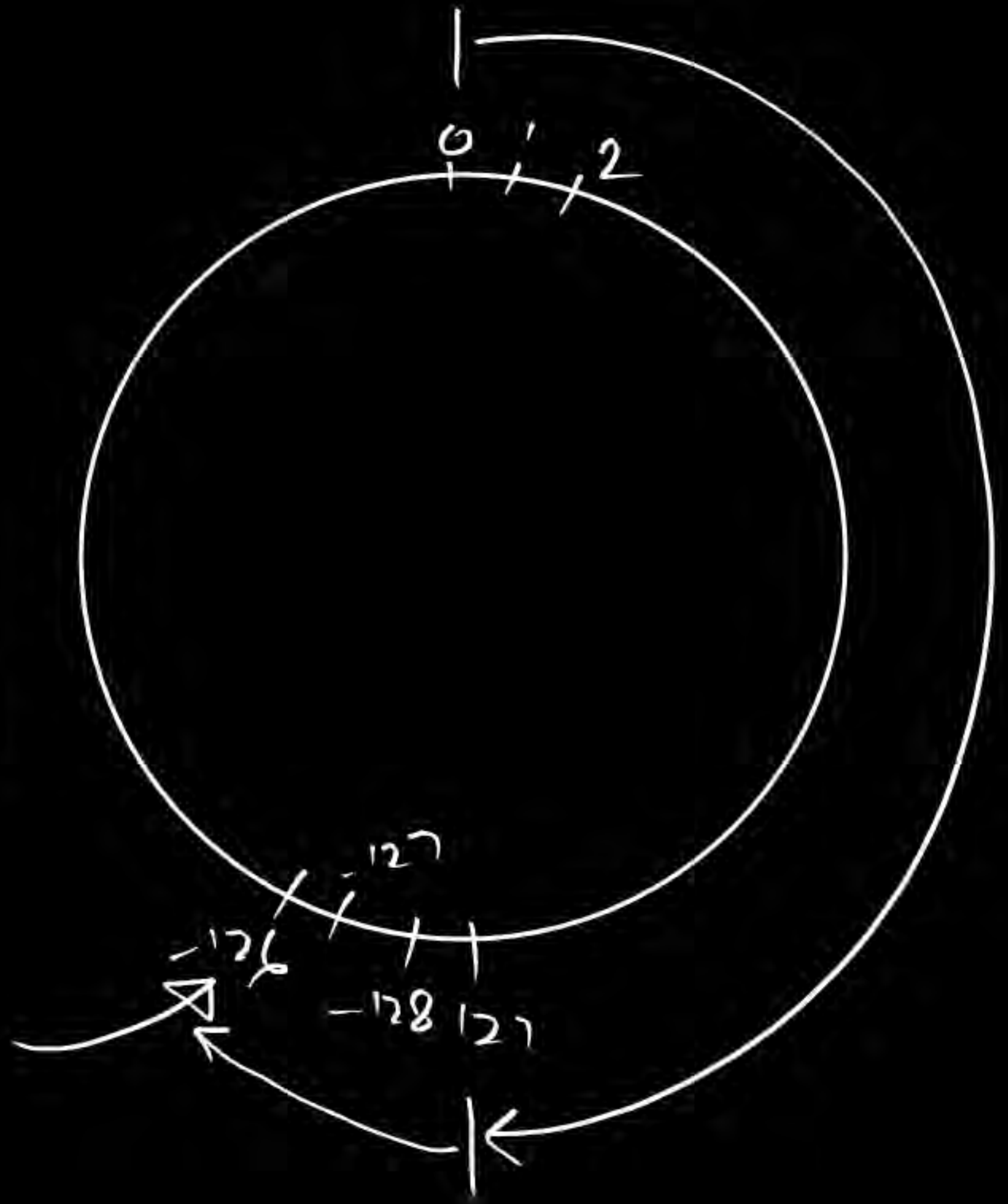
00000000	0000 0000	00000001	10010000

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

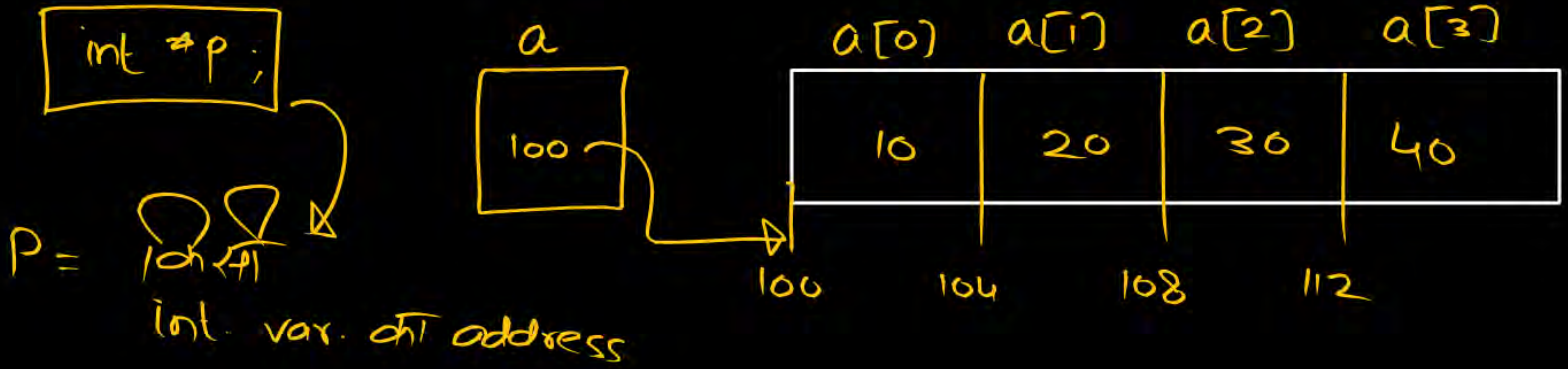
int

```
p= (char*)&x;
```

-126



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



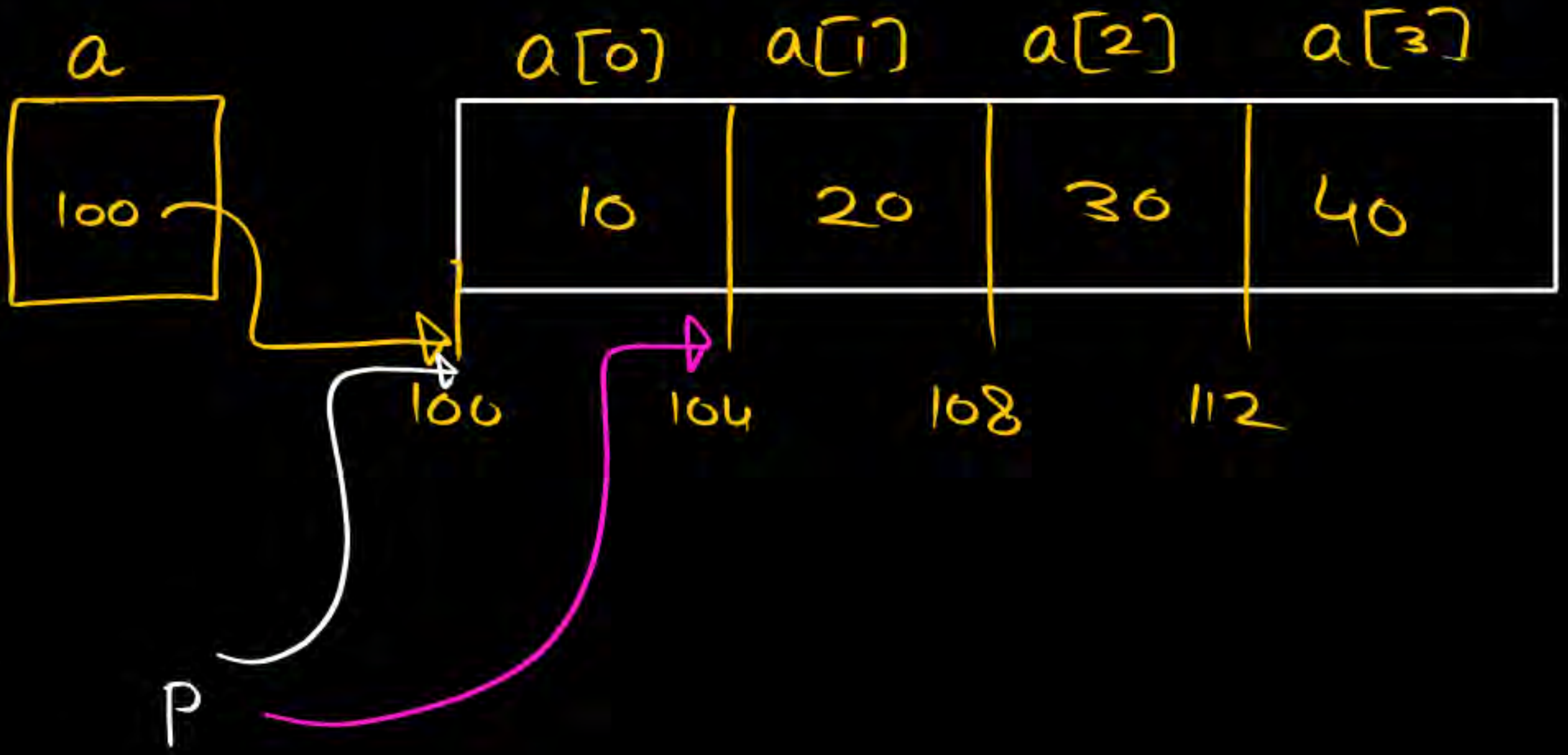
```
int a[4] = {10,20,30,40};
```

```
int *p;
```

```
p = &a[0];
```

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

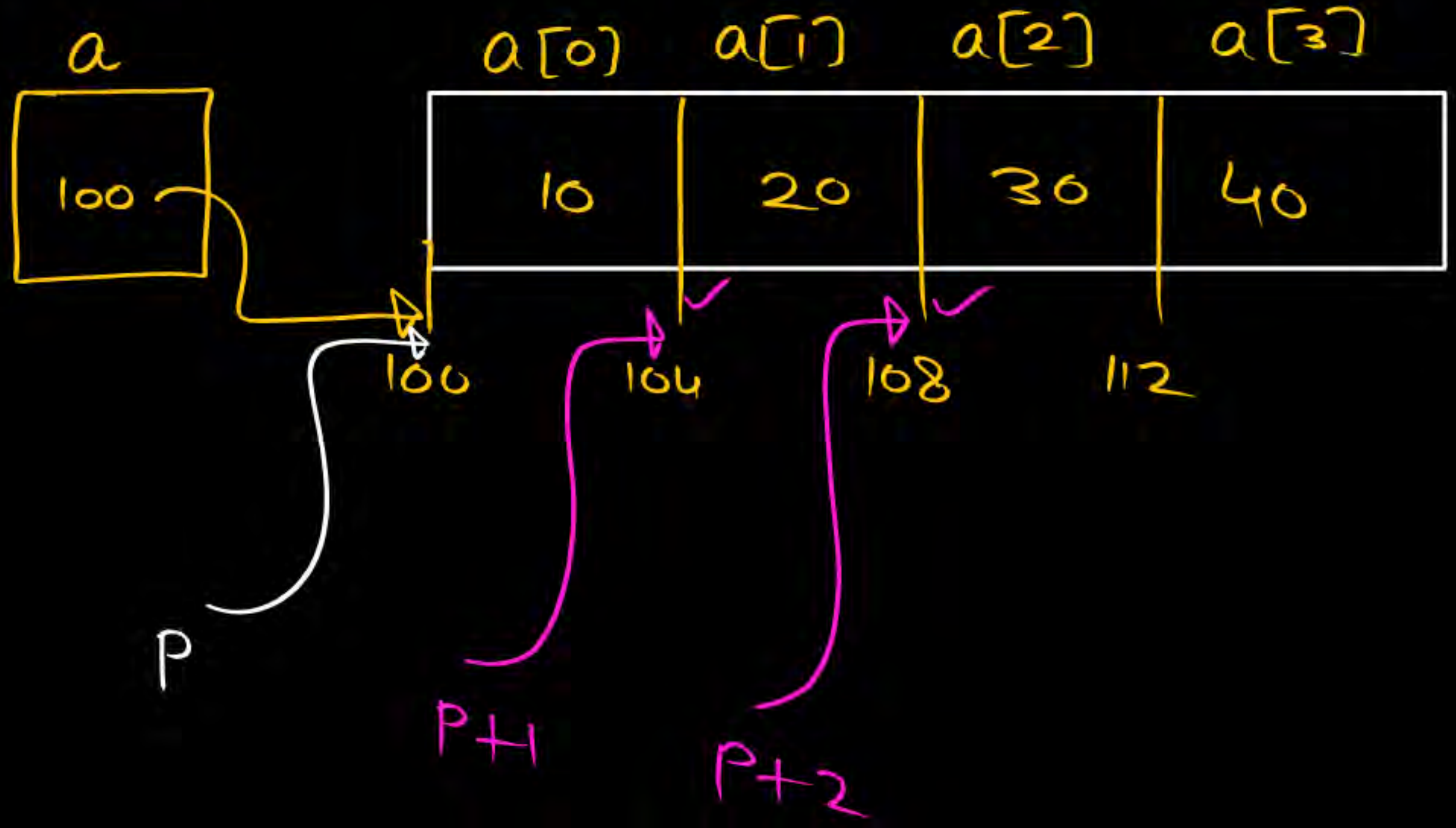
```
p = 100 + 1x4
 p = 104
```



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

```
int *p;
```

```
P = &a[0];
```

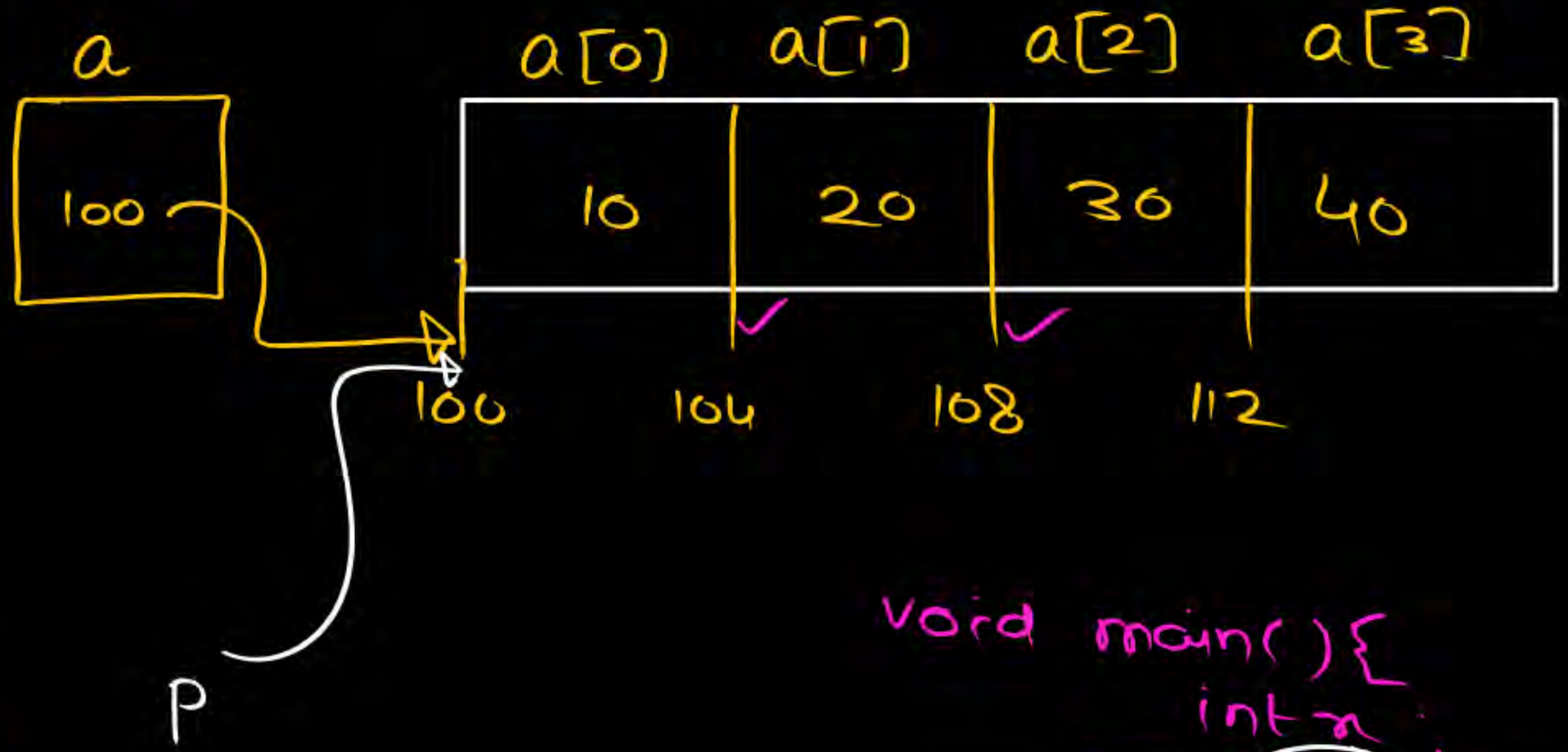


P+1

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

int *p;

p = &a[0];



(i) p+1 ;

Vs

(ii) p = p+1 ;

p+1 → Not updating p

p = p+1 → update p

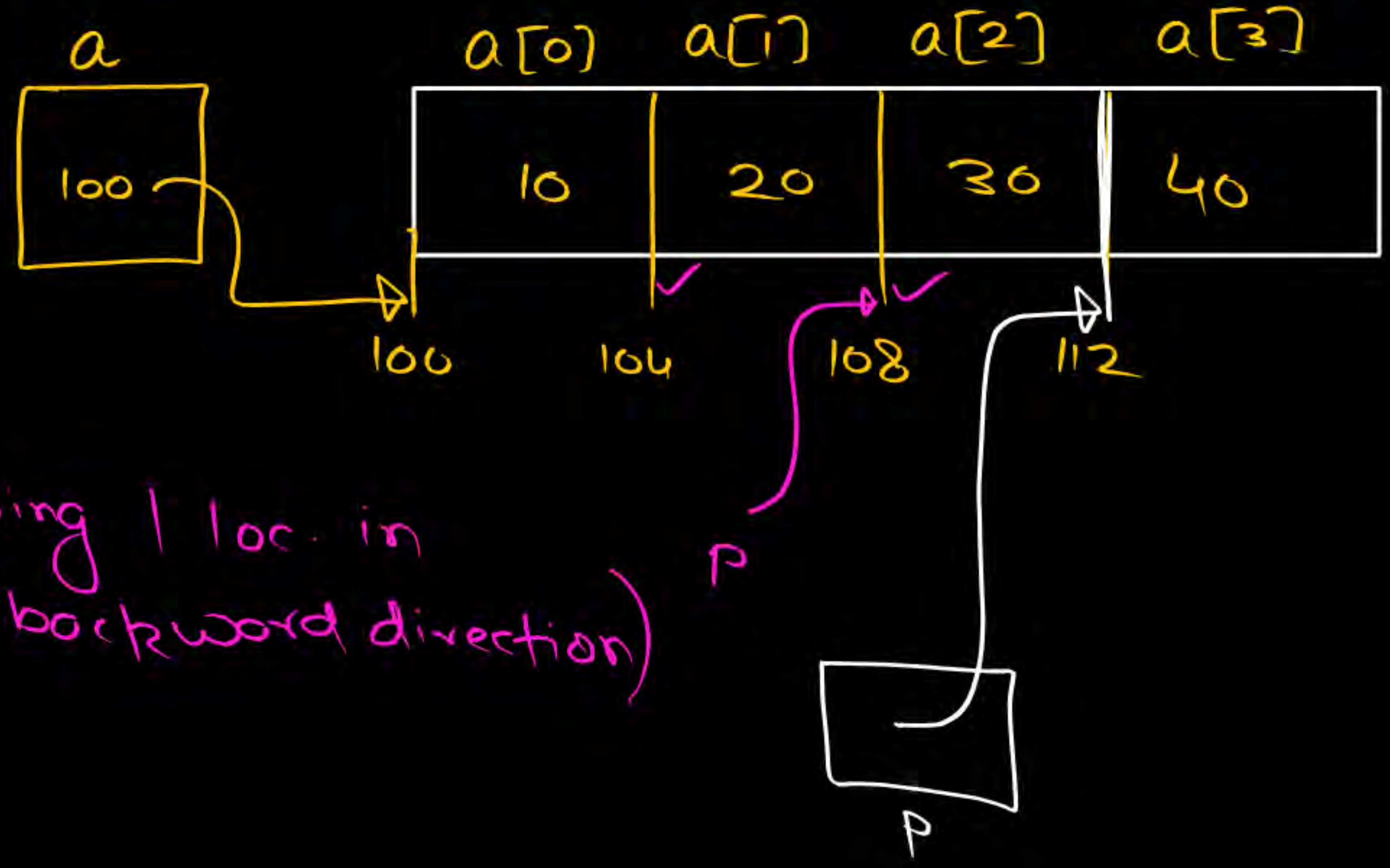
```
void main(){
  int x;
  x = 12;
}
```

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

```
int *p;
```

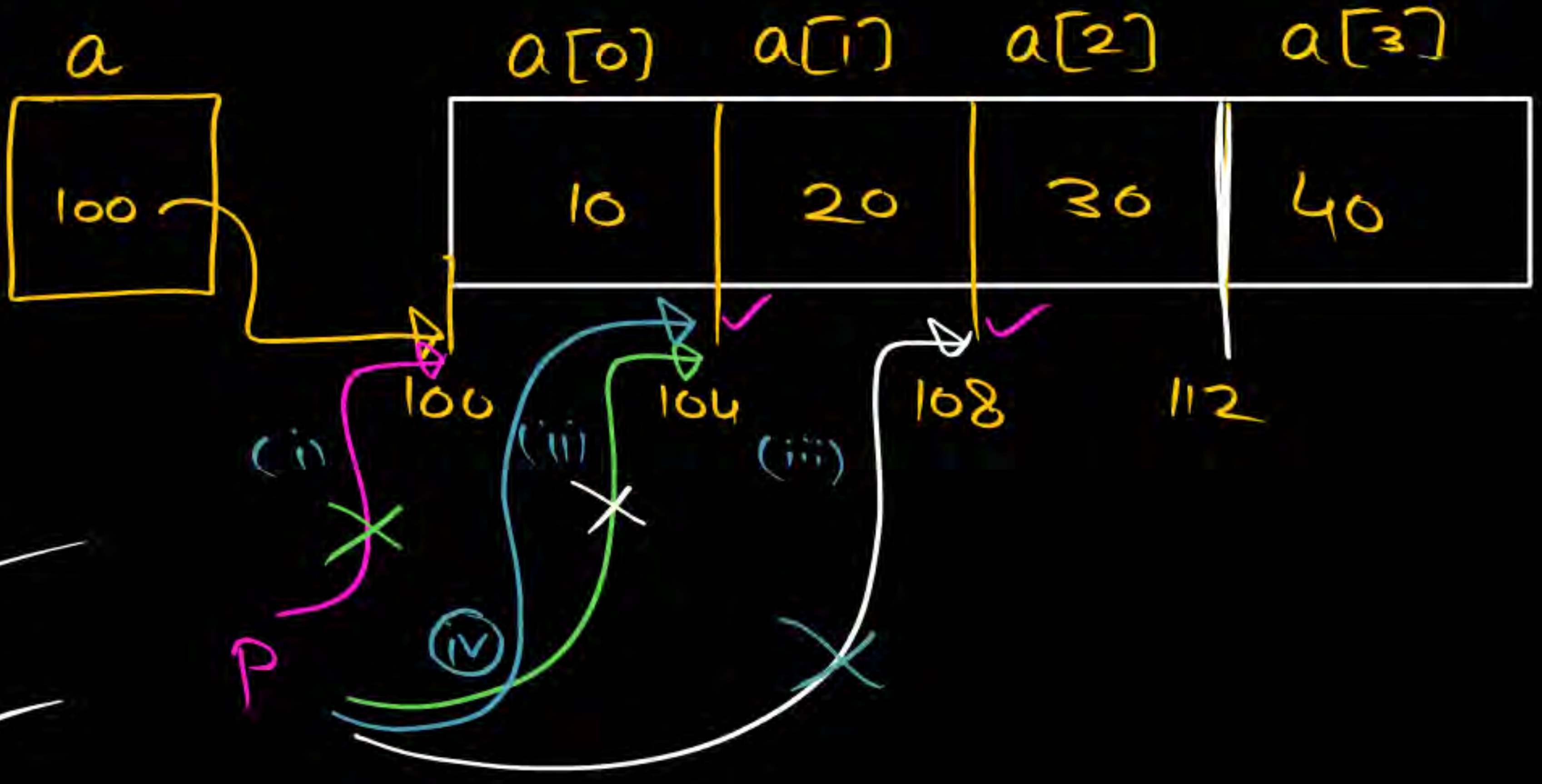
```
p = &a[3]

p = p-1;
```

(moving 1 loc. in backword direction)



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

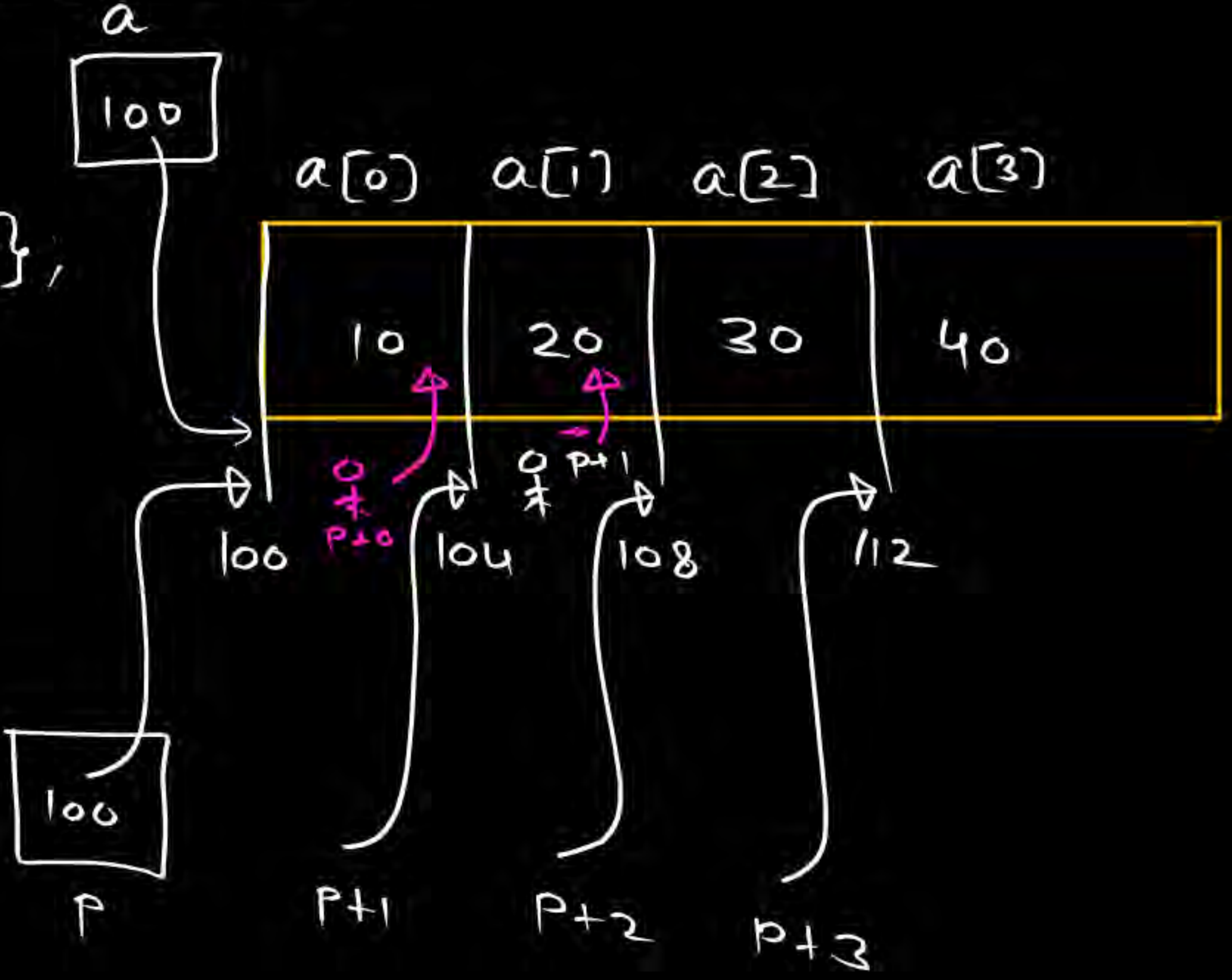
```
int *p;
```



```
p = &a[0];

++p;  ✓

++p;  ✓

--p;  ✓
```

Arrays and Pointers

```
void main(){
    int a[4]={10,20,30,40};
    int *p;
    p = a  OR  p=&a[0];

✓ 10        pf("%d", *(p+0));
✓ 20        pf("%d", *(p+1));
✓ 30        pf("%d", *(p+2));
✓ 40        pf("%d", *(p+3));
```

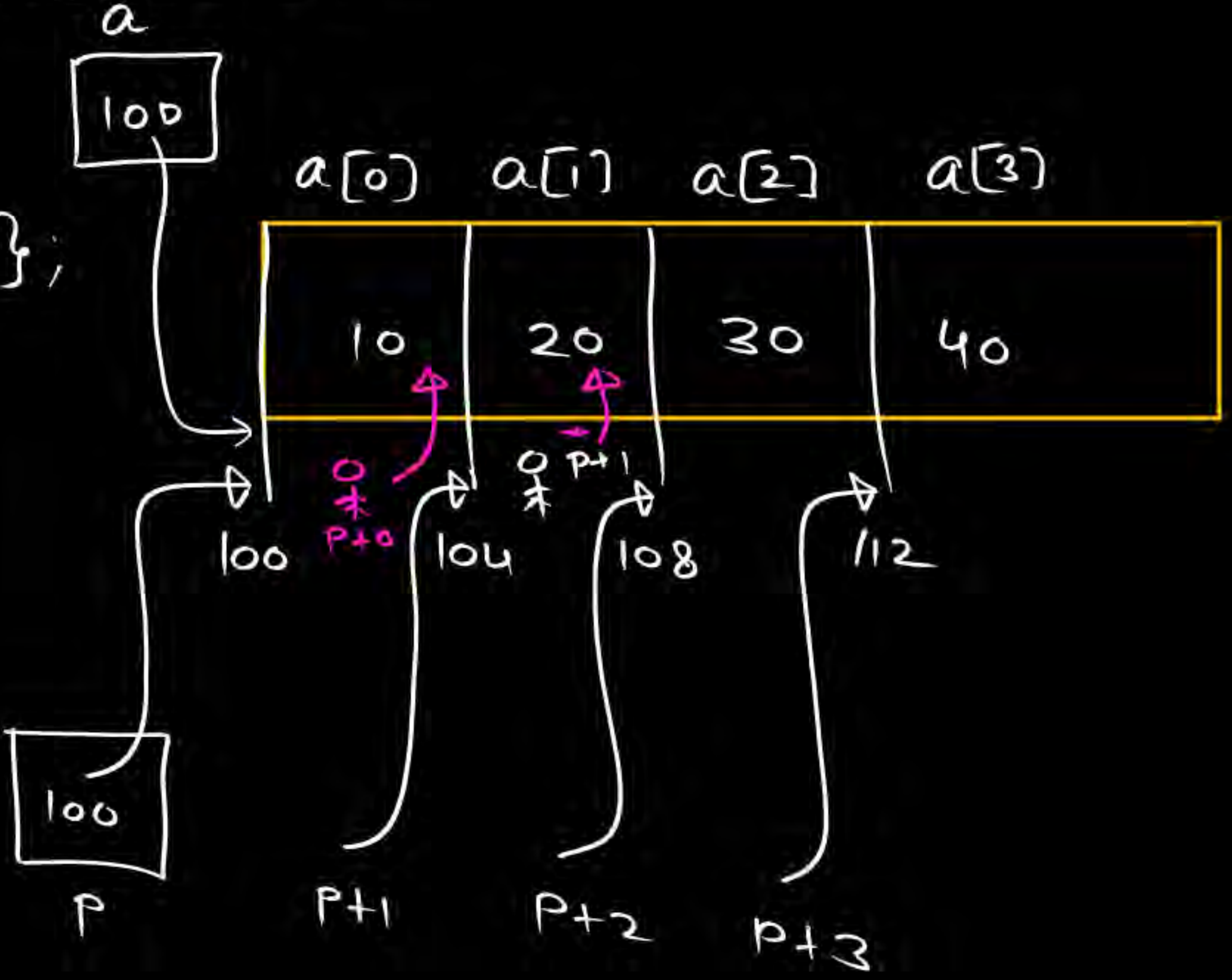


Arrays and Pointers

```
void main(){
    int a[4]={10,20,30,40};
    int *p ;
    P = a  OR  P = &a[0];

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

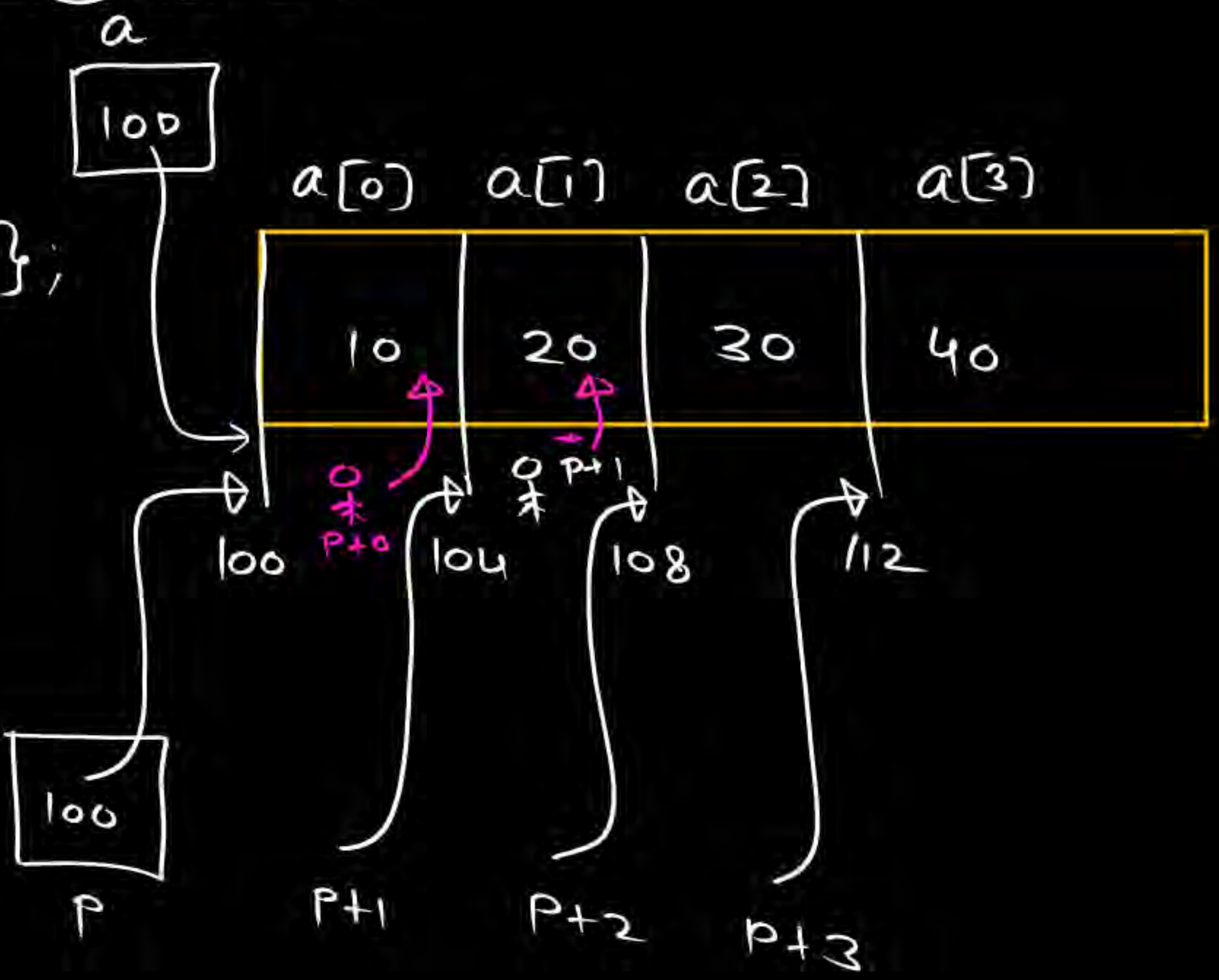
$*(a+i)$

$= a[i]$



Arrays and Pointers

```
void main(){
    int a[4]={10,20,30,40};
    int *p ;
   ┌──────────────────────────────┐
   │ P = a   OR   P = &a[0];      │
   │                              │
   └──────────────────────────────┘

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



What is p → array
→

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

①

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

Invalid

②

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

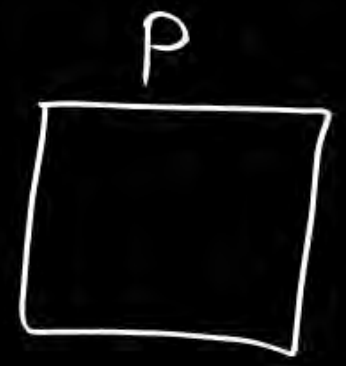
Invalid

array name can not be Lvalue

①

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

✓



②

```
p = &a[2];
```

✓

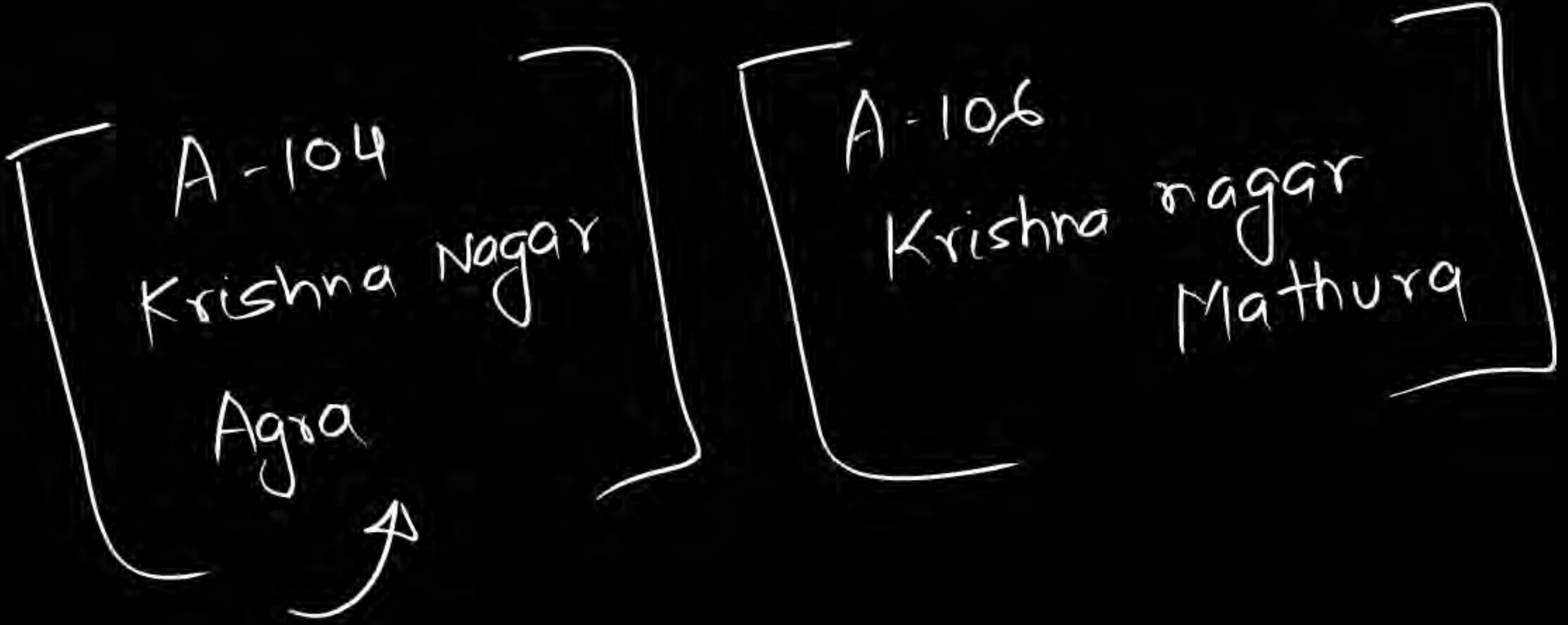
variable =

Pointer_var ++

Pointer_var --

++Pointer_var

--Pointer_var



A-104
Krishna Nagar
Agra

A-106
Krishna nagar
Mathura

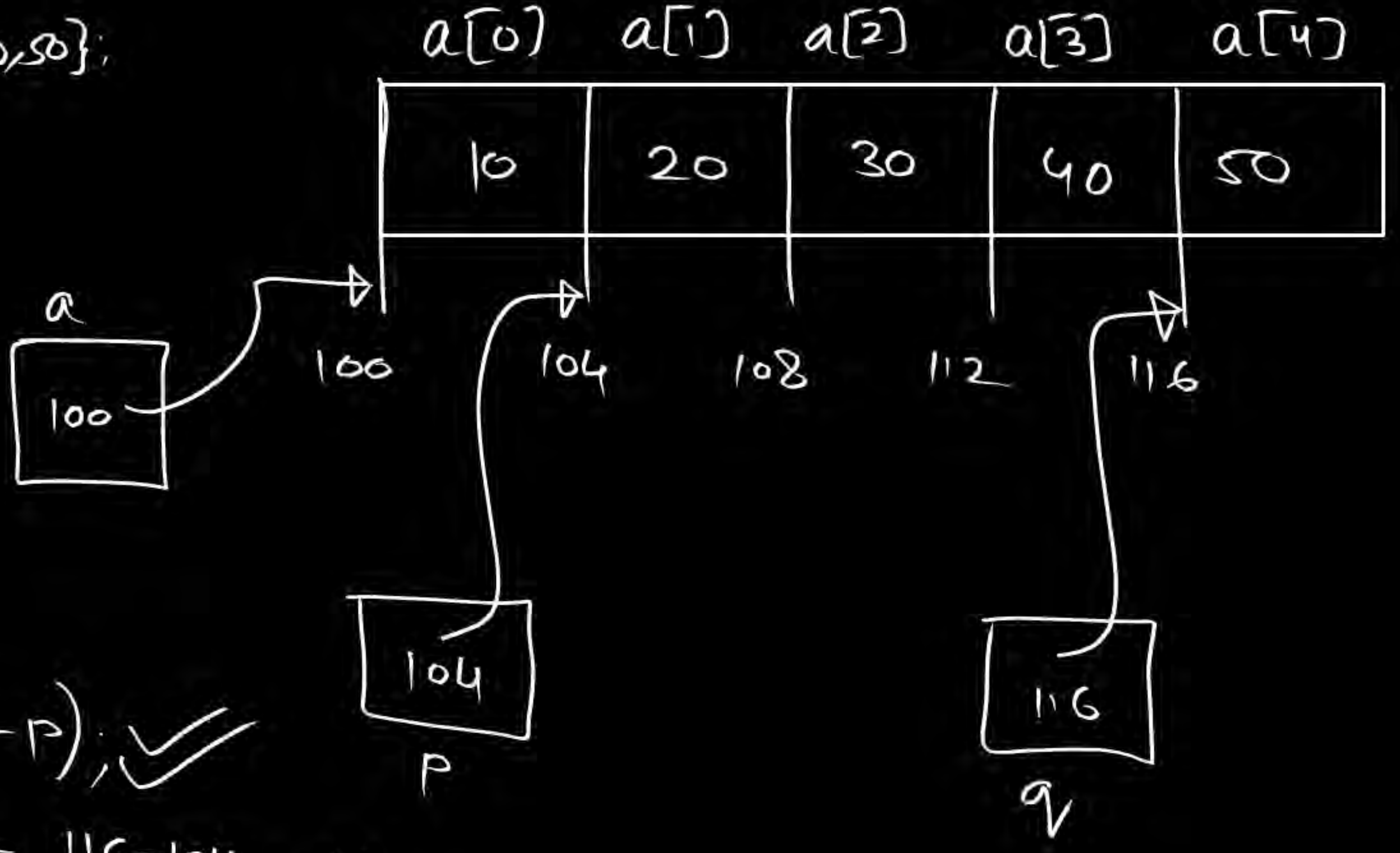
Pointer_var + 3 ⇒ Moving 3 location in forward direction

Pointer_var - 3 ⇒ " " " " backward "

$P_1 + P_2$ ✗

difference →

```
int a[5] = {10,20,30,40,50};

int *p, *q;

p = &a[1];
q = &a[4];
```



```
pf("%d", q-p);
```
✓

$$\frac{\text{actual diff}}{\text{data size}} = \frac{116-104}{4} = \frac{12}{4} = 3$$

THANK YOU GW SOLDIERS !