CS & IT ENGINEERING

C Programming

Arrays and Pointers

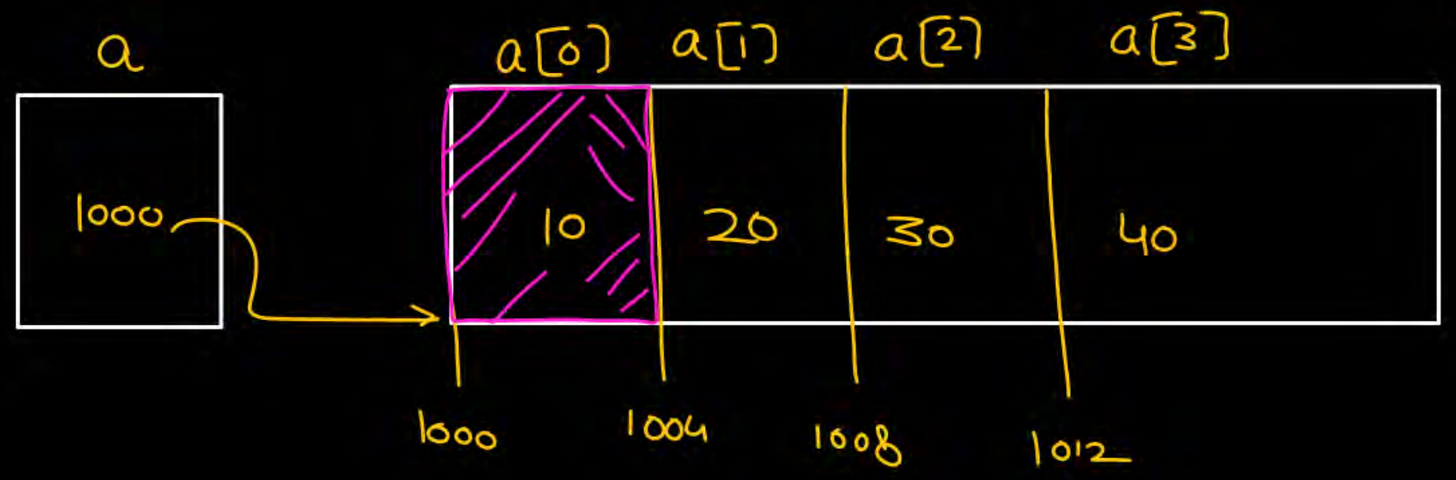
Lec - 02

By- Pankaj Sharma Sir

- Arrays and Pointers (Part- 02)

1) Array-name represents address of its first element.

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




```
1000 {  printf("%u", a);
        printf("%u", &a[0]);
```

$a = \&a[0]$

2) Array name does not represent an address with 2 operators

(i) & : address of operator

(ii) sizeof() operator

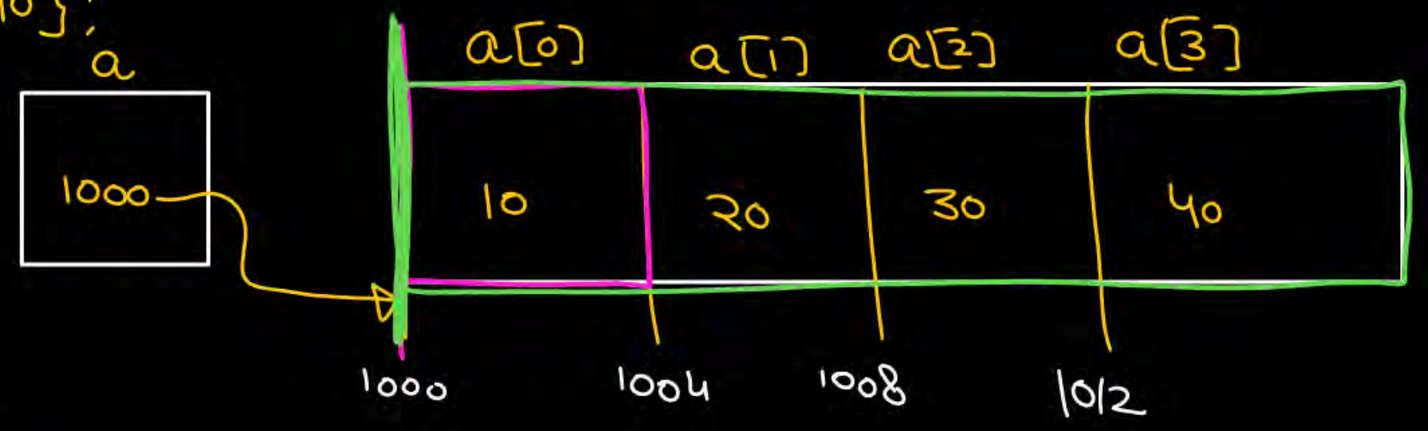
```
int a[4] = {10,20,30,40};
```

4 byte के element के address

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

→ 1000

address of whole array
1 byte के element



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

Numerical value is same but logically we are talking about diff. element's address.

[4 byte
16 byte]

③

Numerical value of a is 100

what is

$\boxed{a+1}$

a is simple var. → value

a is an address / pointer variable → address arithmetic

```
int a = 100;
pf("%d", a+1);
```

or

Maths. value + value = value

address + value = address

value + address = address

address + address ⇒ Invalid

4. If the declaration(Initialization) of an array has n-dimension

int a[4]; 1 dim

int a[2][3]; 2 dim

int a[2][3][4]; 3 dim

a) Anywhere other than declaration if we provide exactly n-dim. then we are working on element.

b) if we provide less than n-dim then we are working on addresses.

int a[4]; 1-dim

≡

a ⇒ add/element (0-dim)

→ address

a[1] // 1-dim

→ add ✗

→ element ✓

int a[2][3]; 2-dim.

=

a → 0-dim → address ✓

→ element

a[0] → 1-dim → add ✓

→ element ✗

a[1] → 1-dim → add ✓

→ element ✗

a[0][0] → 2-dim → add ✗

→ element ✓

$a + 1$?

what is a

a

Value

dance करेंगे

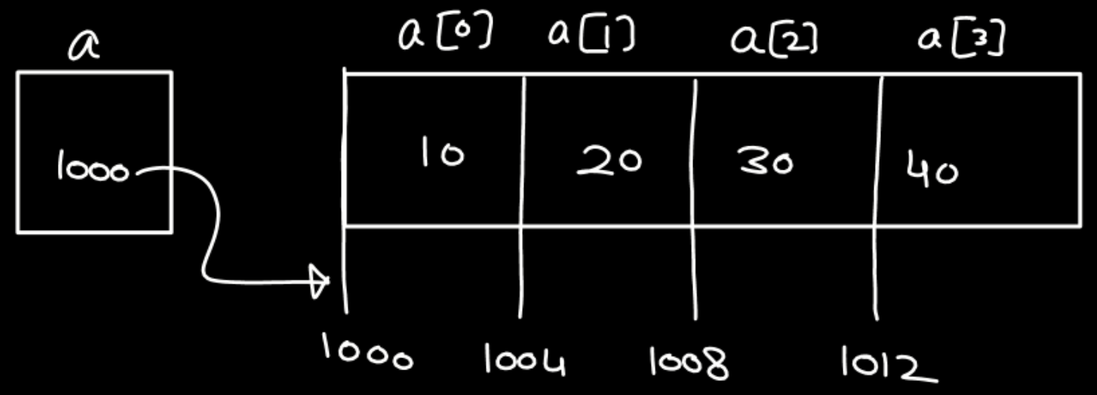
a is address

(i) किसका address है ?

(ii) उसका size क्या है ?

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

printf("%u", a+1);
```

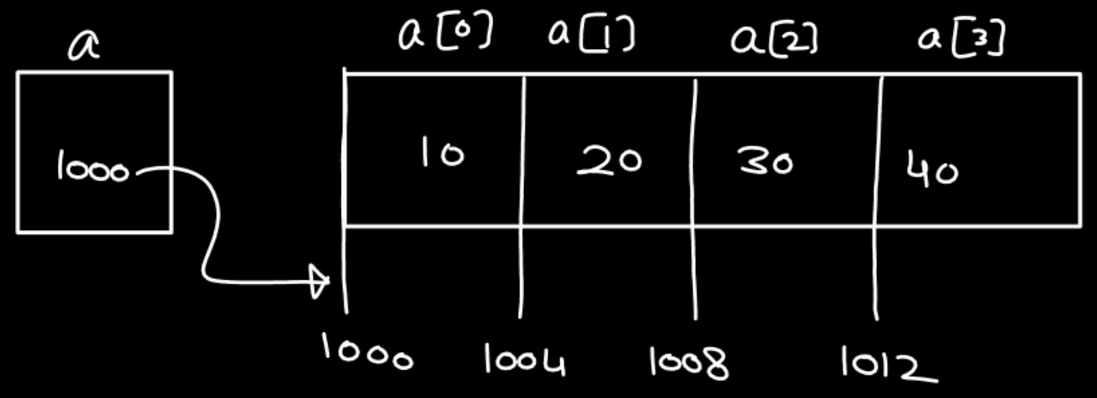


1004

a+1
- (i) a → address ✓
  - → element ✗
- (ii) किसका address है ? ⇒ a is array name which rep. add. of its 1st elem. ⇒ &a[0]
- (iii) What is size of a[0] ⇒ 4 byte

$$a+1 = \&a[0] + 1$$
$$= \&a[0] + 1 \times 4$$
$$= 1000 + 4 \Rightarrow 1004$$

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



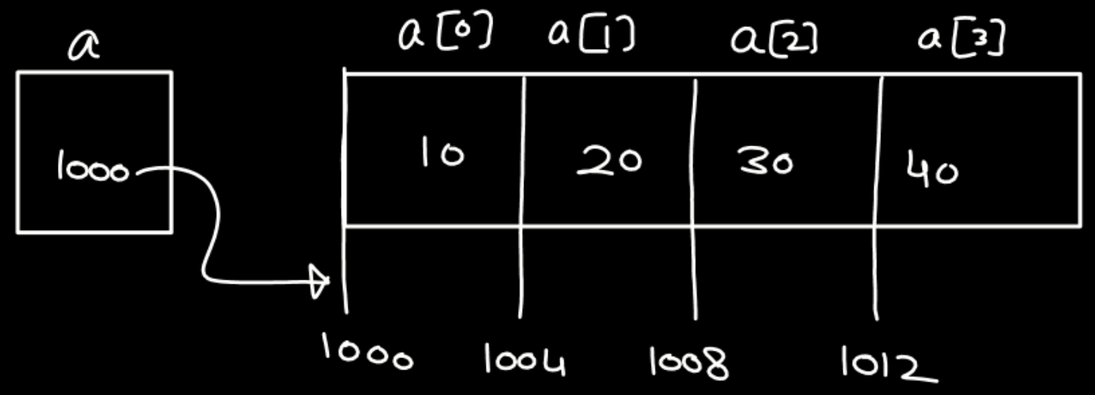
$$a+1 = \&a[0] + 1$$
$$= \&a[0] + 1 \times 4 = 1004$$

$$a+2 = \&a[0] + 2 \times 4$$
$$= 1008$$

$$\boxed{(a+1) = \underset{1004}{\text{Memory location}} = \&a[1]}$$

$$\boxed{(a+2) = \underset{1008}{\text{Memory location}} = \&a[2]}$$

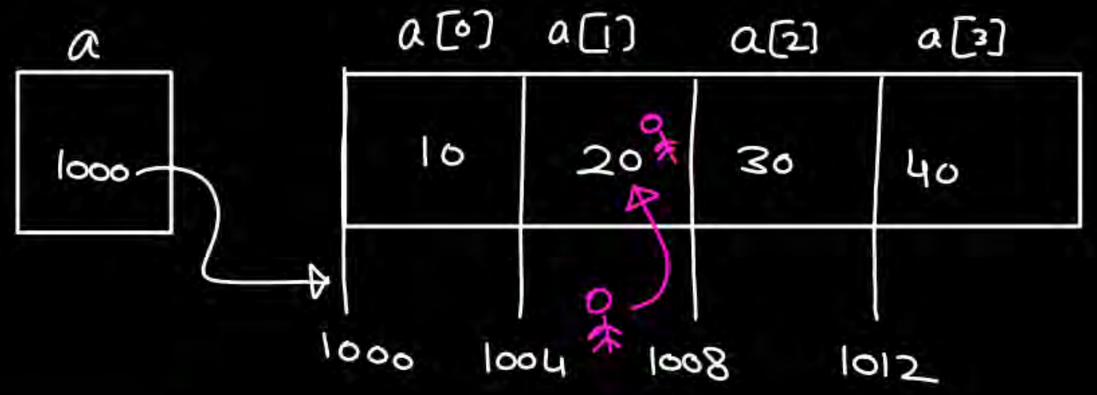
int a[4] = {10, 20, 30, 40};



$$\boxed{(a+1) = \text{Memory location } 1004 = \&a[1]} \quad -(1)$$

$$\boxed{(a+2) = \text{Memory location } 1008 = \&a[2]} \quad -(2)$$

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



$(a+1)$ = Memory location 1004 = &a[1] —(1)

$(a+2)$ = Memory location 1008 = &a[2] —(2)

(a+2) = value at (Memory location 1008) = ~~~~ ~~&~~a[2]

*(a+2) = a[2]

(a+1) = value at (Memory location 1004) = ~~~~ ~~&~~a[1]

*(a+1) = a[1]

*(a+2) = a[2]

*(a+i) = a[i]

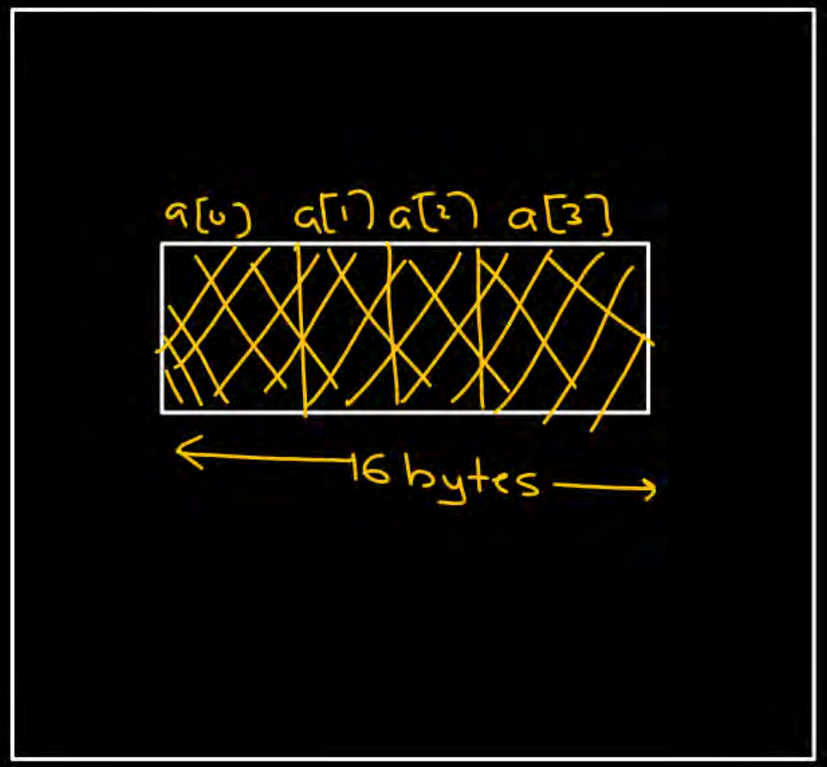
```
int a[4] = {1, 2, 3, 4};
```

```
a[4] = 100;
```

→ behaviour is undefined

*(value)

⟹



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

pf("%d", a[1]);
pf("%d", *(a+1));
```

$\Rightarrow$ same

Addition is commutative

$$2+3 = 3+2$$

$$\boxed{a[1] = *(a+1) = *(1+a) = 1[a]}$$

```
int a[4] = {10,20,30,40};
pf("%d", a[2]);
pf("%d", *(a+2));
pf("%d", 2[a]);
pf("%d", *(2+a));
```

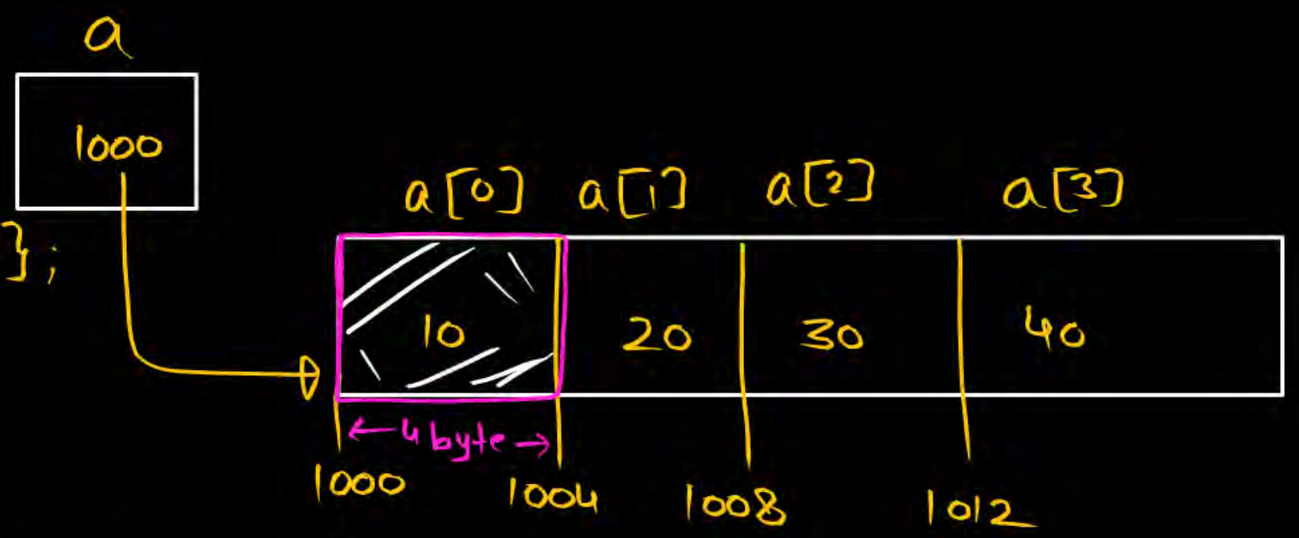
30

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

→ Error

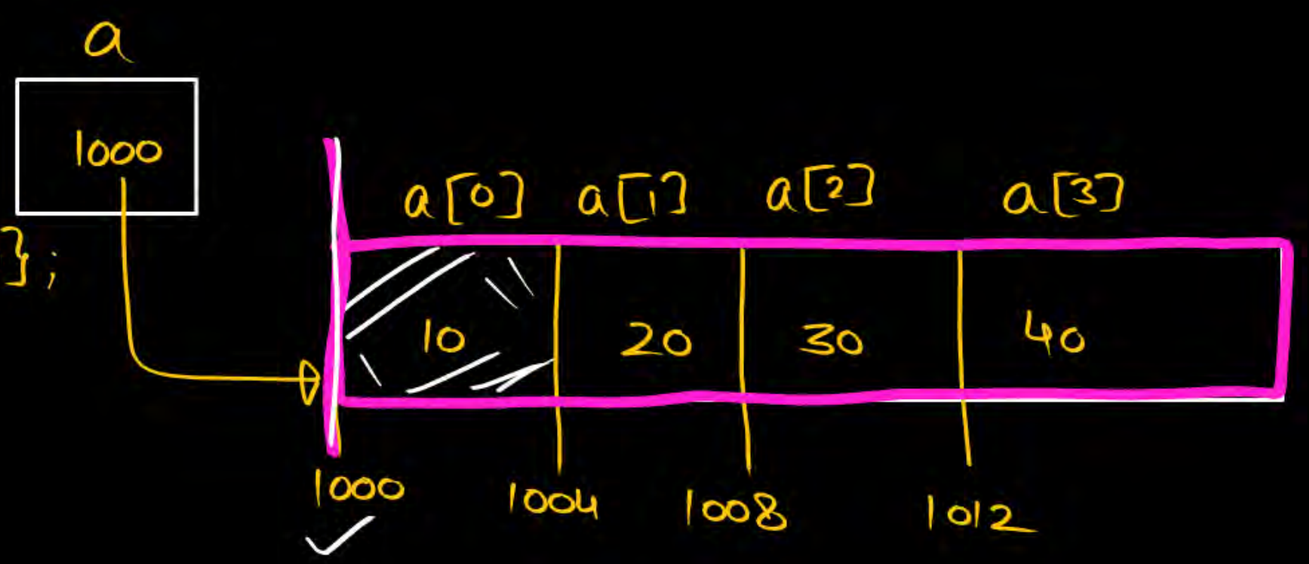
$$\boxed{a[i] = *(a+i) = *(i+a) = i[a]}$$

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



array का address ⇒ first ele. add = &a[0]
= 1000

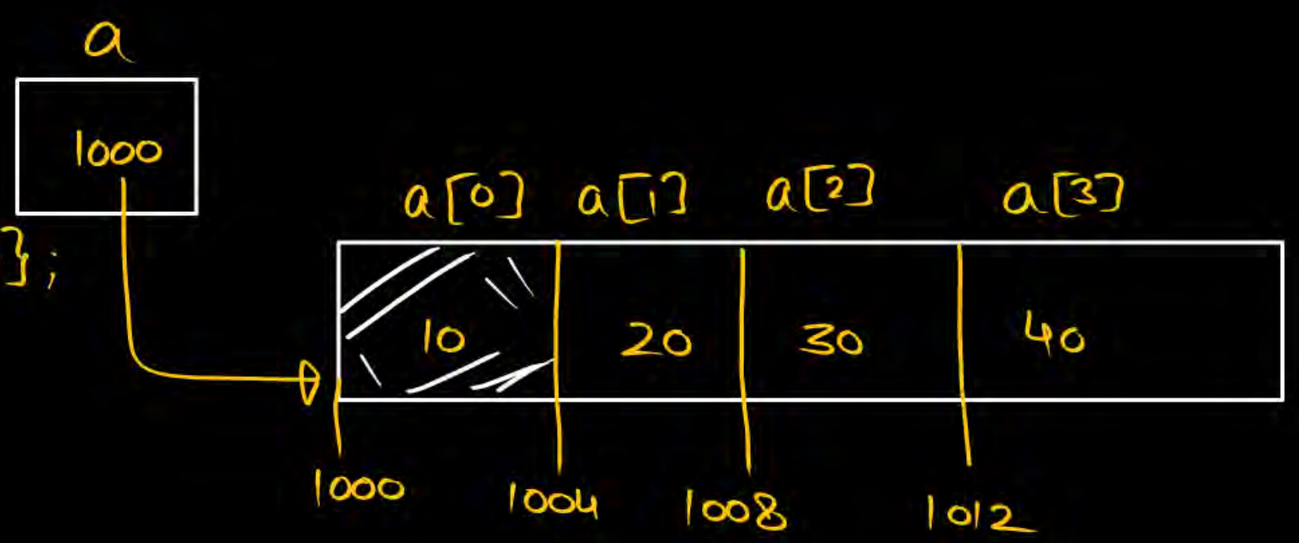
```
void main(){
    int a[4] = {10,20,30,40};
1000  printf("%d", a);
1000  printf("%d", &a);
      printf("%d", a+1);
      printf("%d", &a+1);
    }
```



पूरे array का address ⇒ 1000

(16 byte के elem.)

```
void main(){
    int a[4] = {10,20,30,40};
1000  printf("%d", a);
1000  printf("%d", &a);
1004  printf("%d", a+1);
      printf("%d", &a+1);
    }
```



a+1 ⇒ a being array name ⇒ उसके first ele. का address

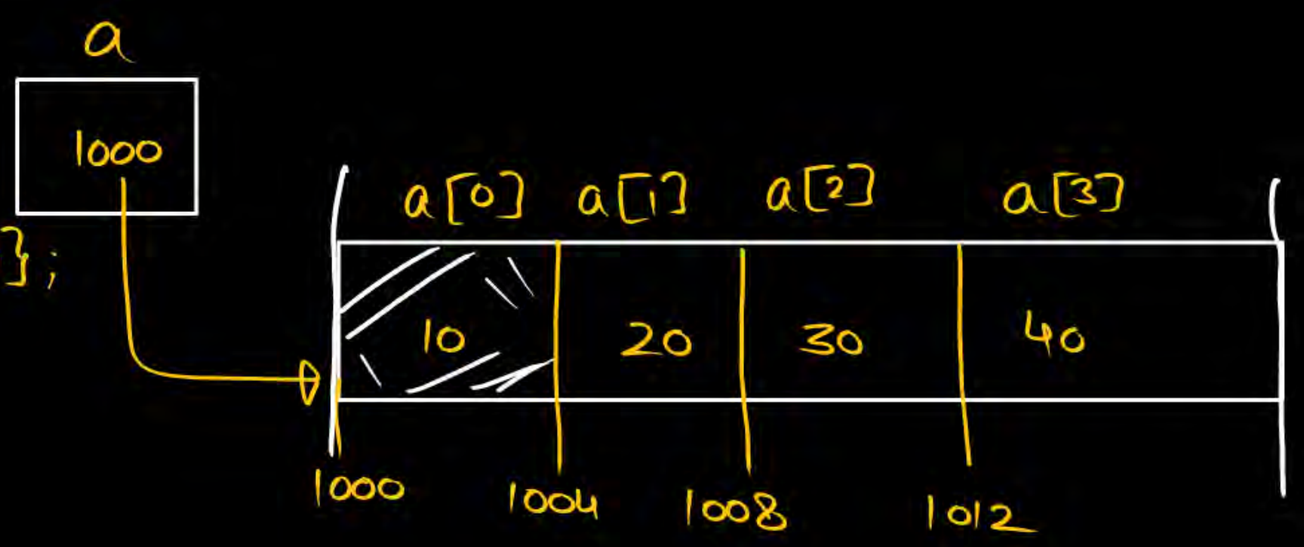
$a = \&a[0]$

$a+1 = \&a[0] + 1$ (a[0] → 4 byte)

$1000 + 1 \times 4$

$= 1000$

```
void main(){
    int a[4] = {10,20,30,40};
1000  printf("%d", a);
1000  printf("%d", &a);
1004  printf("%d", a+1);
      printf("%d", &a+1);
    }
```



$\&a + 1 \Rightarrow \&a + 1 \times 16 = 1000 + 16$

$= 1016$

पूरे array का address

पूरे array का size (16 byte)

Doubts ?

a[1] ⇒ square brackets

Relative addressing

*(a+1)

address +1 ⇒ *(add)

6 hrs

$l\,a[0] + 1$

Constant

$$\left\{\begin{array}{l} 300x + 200y = 30 \\ 200x - 400y = 40 \end{array}\right\}$$

$\Rightarrow$

1 small square = 10 units


THANK YOU GW
SOLDIERS !