CS & IT ENGINEERING

C Programming

Arrays and Pointers

Lec - 01

By- Pankaj Sharma Sir

- Arrays and Pointers (Part- 01)

Arrays & Pointers

① Address → Abs. address

→ Relative address

Abs. address

A-106 Krishna Nagar

Mathura - 281004 (UP)

A-101	A-102	A-103	A-104	A-105	A-106

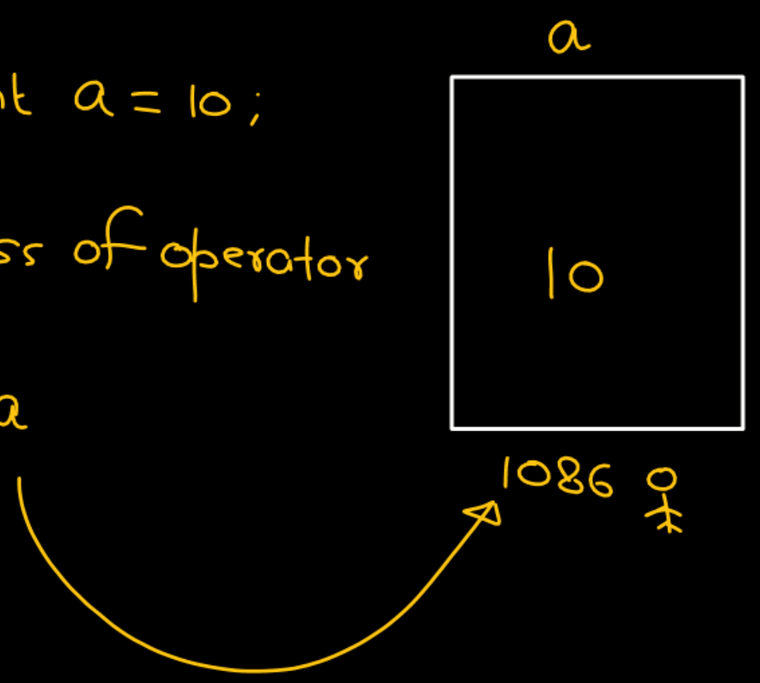
Knock-Knock

② How to find address of some var.

int a = 10;

& : address of operator

&a

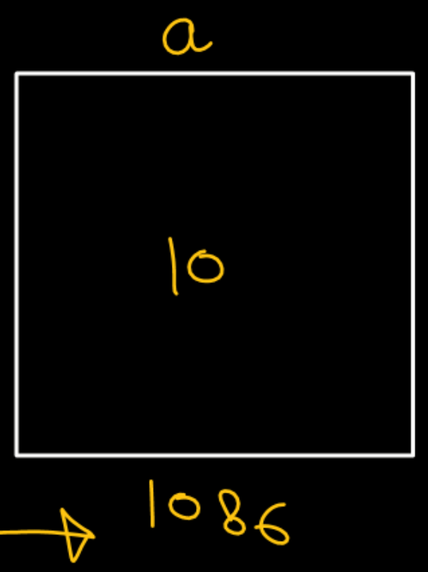


③ * : value at operator

value at (Some Memory location)

int a = 10;

&a : Memory location 1086



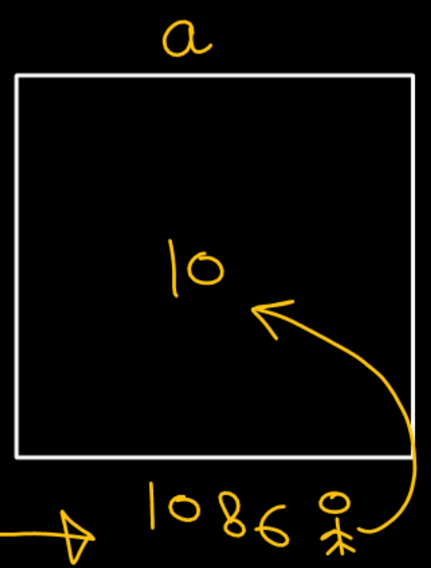
*(&a) ⇒ value_at (Memory location 1086) = 10

③ * : value at operator

value at (Some Memory location)

int a = 10;

&a : Memory location 1086



*(&a) ⇒ value_at (Memory location 1086) = 10

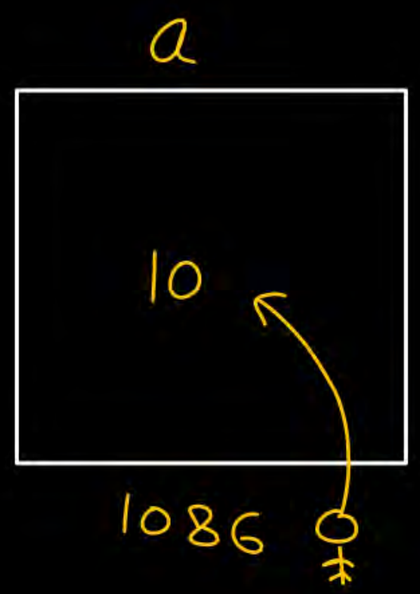
&a = Mem loc. 1086

$\boxed{*(\&a) = 10}$

int a = 10

&a

*(&a)



int a = 10;

pf("%d", *&*&*&a);

$\cancel{*}(\cancel{\&}a) = a$

Why arrays?

```
int m1,m2,m3;
float avg;
scanf("%d%d%d",&m1,&m2,&m3);
avg = (m1+m2+m3)/3.0;
```

multiple var. of same type

int m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12, m13, ...

```
scanf("%d%d%d%d.....", &m1, &m2
                       ...)
```

500 students

int a,b,c;

3 variable

each

3 var. of type

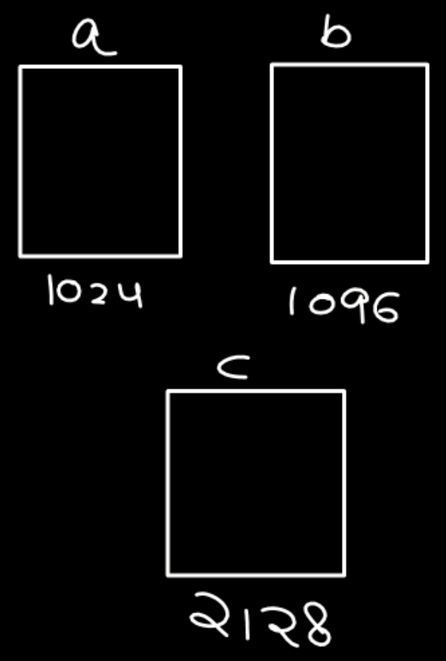
int a[3];

a is a group of

int a,b,c,d,e; ⟹ int b[5];

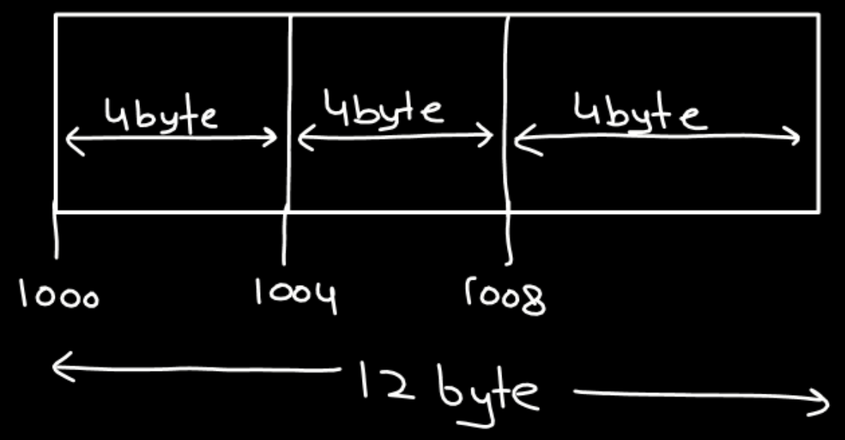
5 var.

int a,b,c;

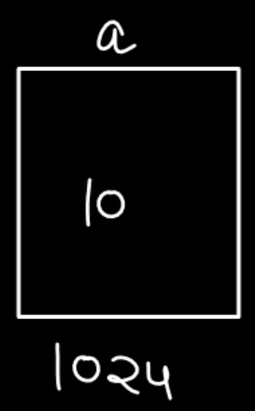


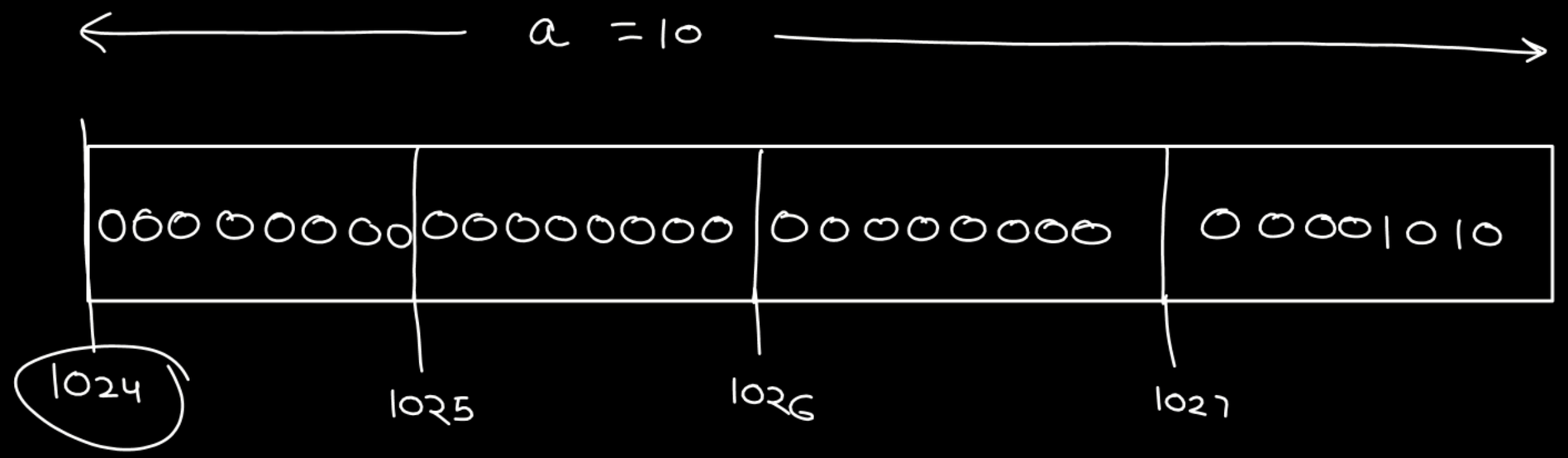
int-4byte

int a[3];



int a = 10;


a
10
1024


a = 10
00000000
00000000
00000000
00001010
1024
1025
1026
1027

```
int a, b, c;

a = 10;
=
b = 12;
=
c = 20;
=
```

```
int a[3];
```

a

3 elements are rep. by same name => a

Sec-A

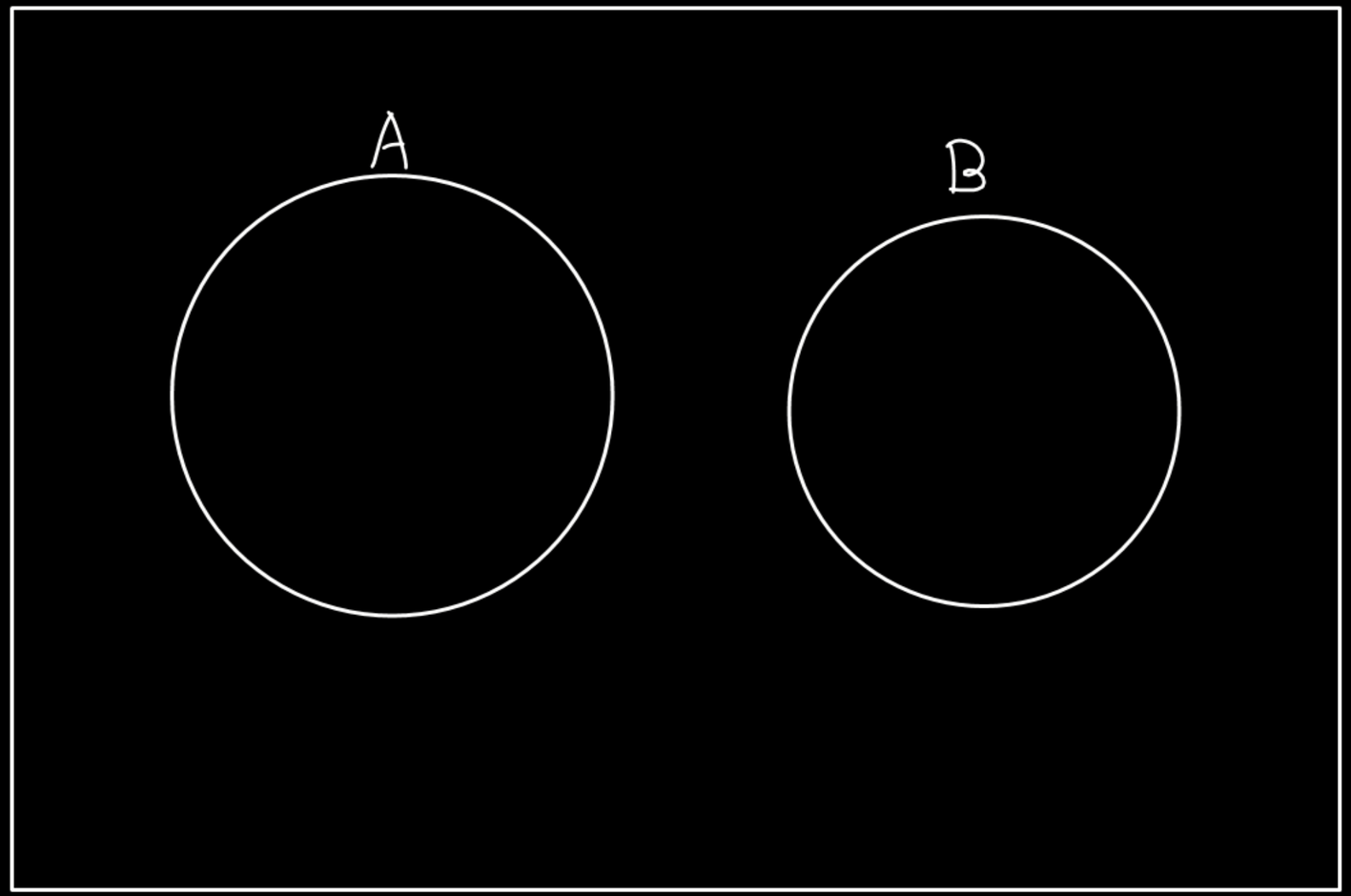
60 students

(group)

akgec

Unique identification

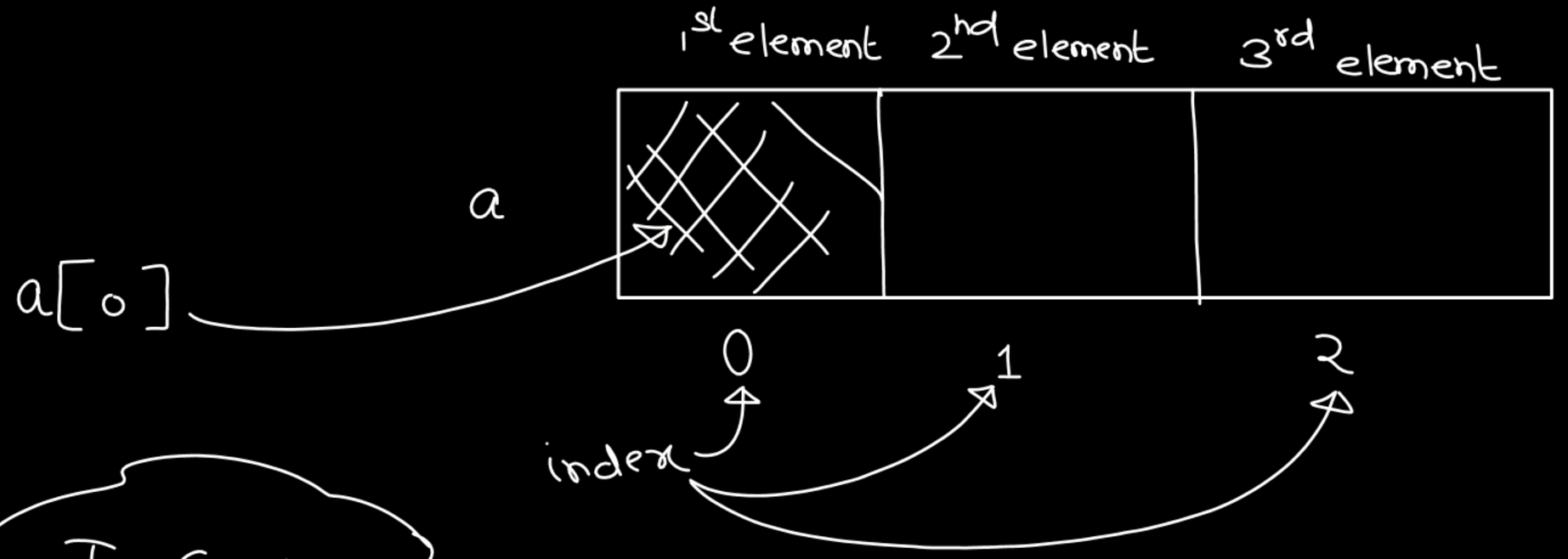
Roll no



A - 1 → unique id no.

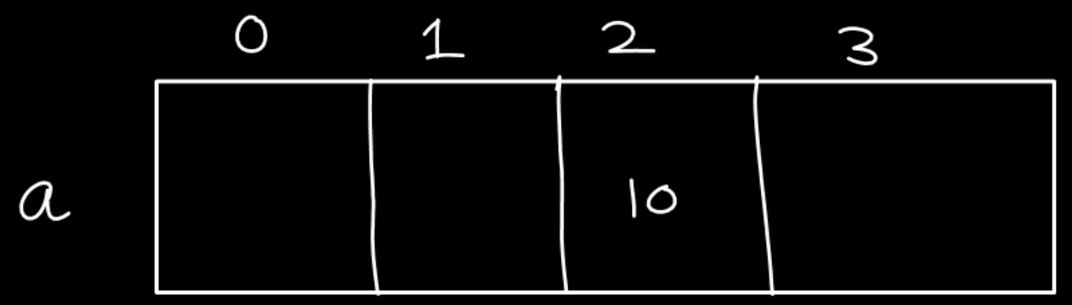
grp name

int a[3];



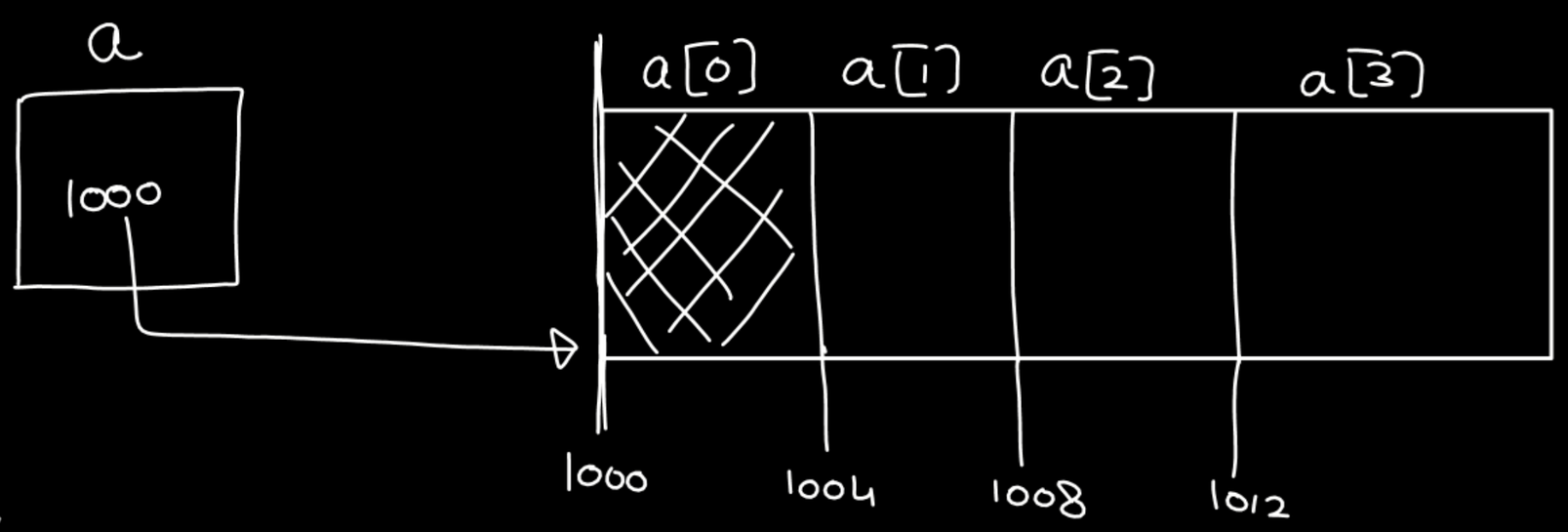
In C, index always starts from 0

```
int a[4];

a[2] = 10;
```



declaration (compiler)

int a[4];



array name ⇒ Constant address

represent address of its first element.

```
++2 ;
--2 ;
2++ ;
2-- ;
```

Invalid

bcz array name is a const. address

```
++ Array_name ;
-- Array_name
Array_name ++
Array_name --
```

Invalid

Lvalue = Rvalue

Lvalue can not be a constant.

Array_name =  Invalid

```
void main(){
    int a;
    printf("%d",a);
}
```

local/
auto

Garbage

```
void main(){
    int a[4];
    printf("%d",a[2]);
}
```

Garbage

a[0]	a[1]	a[2]	a[3]
G	G	G	G

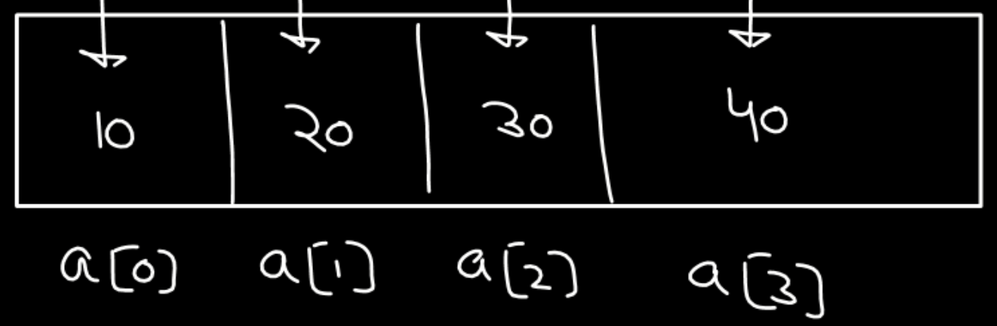
int a[4];

Collection of 4 variable

a[0]
a[1]
a[2]
a[3]

int a = 10; ✓

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



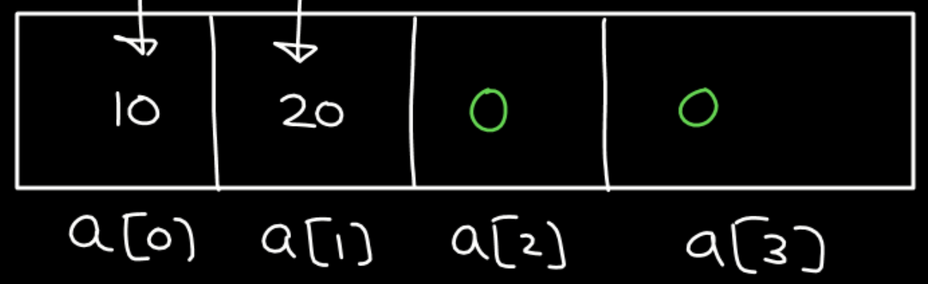
```
int a[4] = {10,20,30,40};

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

int a[ ];
```

Ud fe loat

Marega

Group ⇒ size mention

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



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

3

```
int a[];
```

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

{ }

1 or more values

int a[3]

int a[2]

Variable

```
int a = {10};
int a = 10;
```

```
int a[    ]
```

size

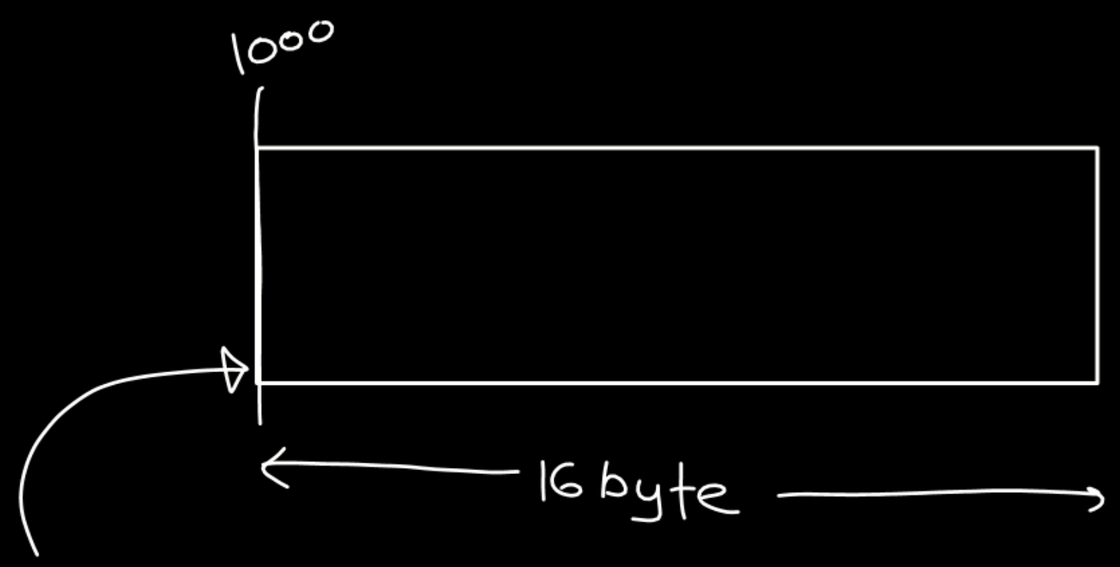
```
#define max 10
void main(){
    int a[max];
    =
}
```

```
int a[10];
int a[2+2x3];
int a[2xsizeof(int)]
```

```
int a[4];
```

16 bytes → cont.

block




THANK YOU GW
SOLDIERS !