unacademy

Arrays & Pointers Part - II

Comprehensive Course on C- Programming

Pankaj Sharma • Lesson 20 • May 27, 2024

unacademy

CS & IT Engineering

C Programming

Arrays & Pointers Part-02

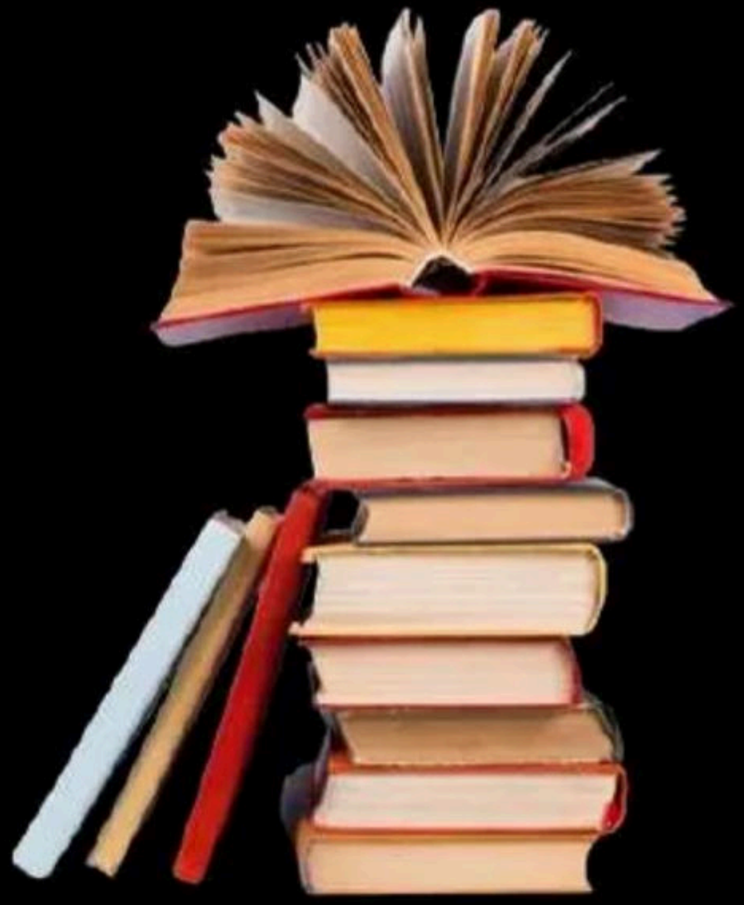
Lecture Number- 20

By- Pankaj Sir

unacademy

Topics

to be covered



1. Arrays & Pointers Part-II

unacademy

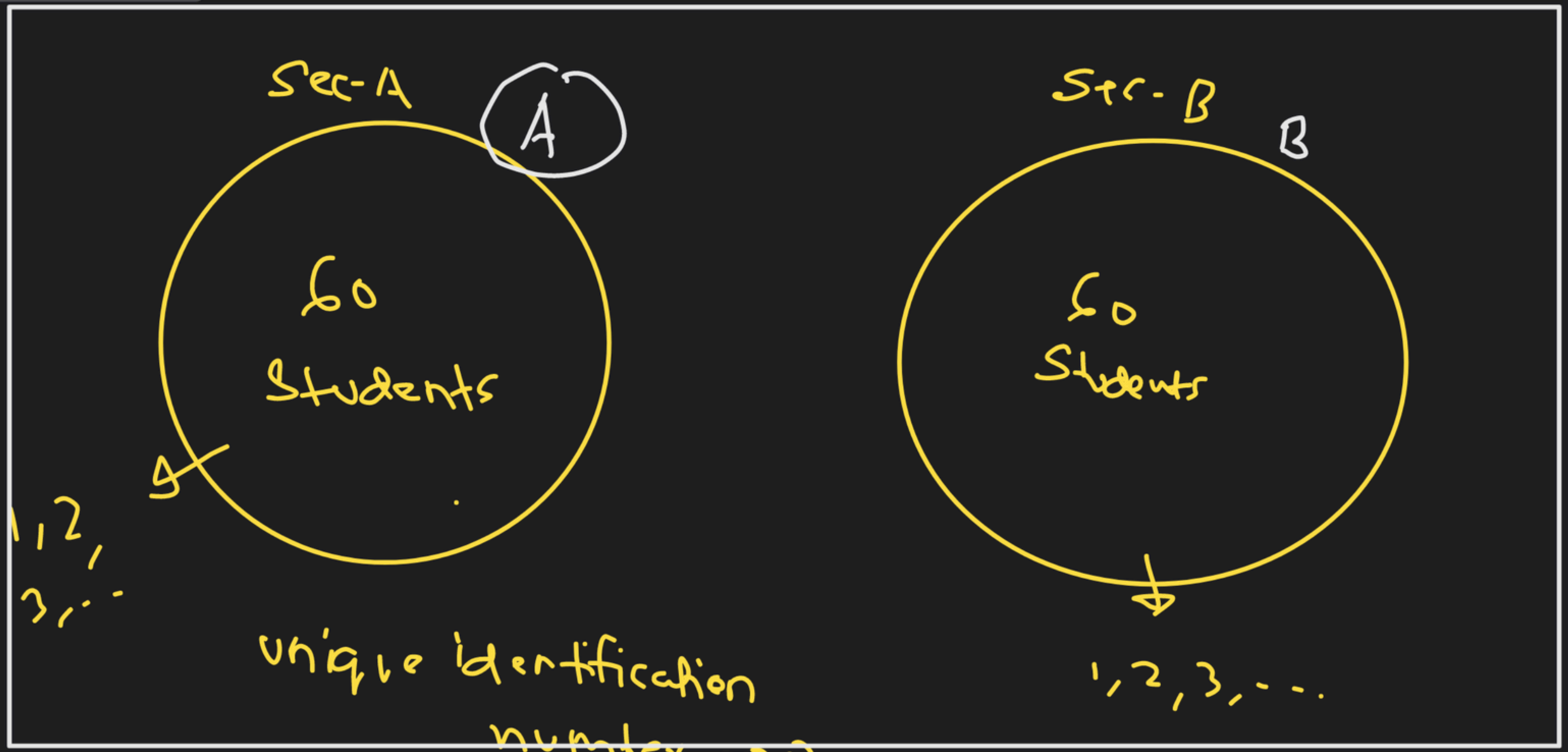
int a, b, c;

a = 10;
≡
b = 30;
≡
c = 300;

int a[3]; Group/collection

a

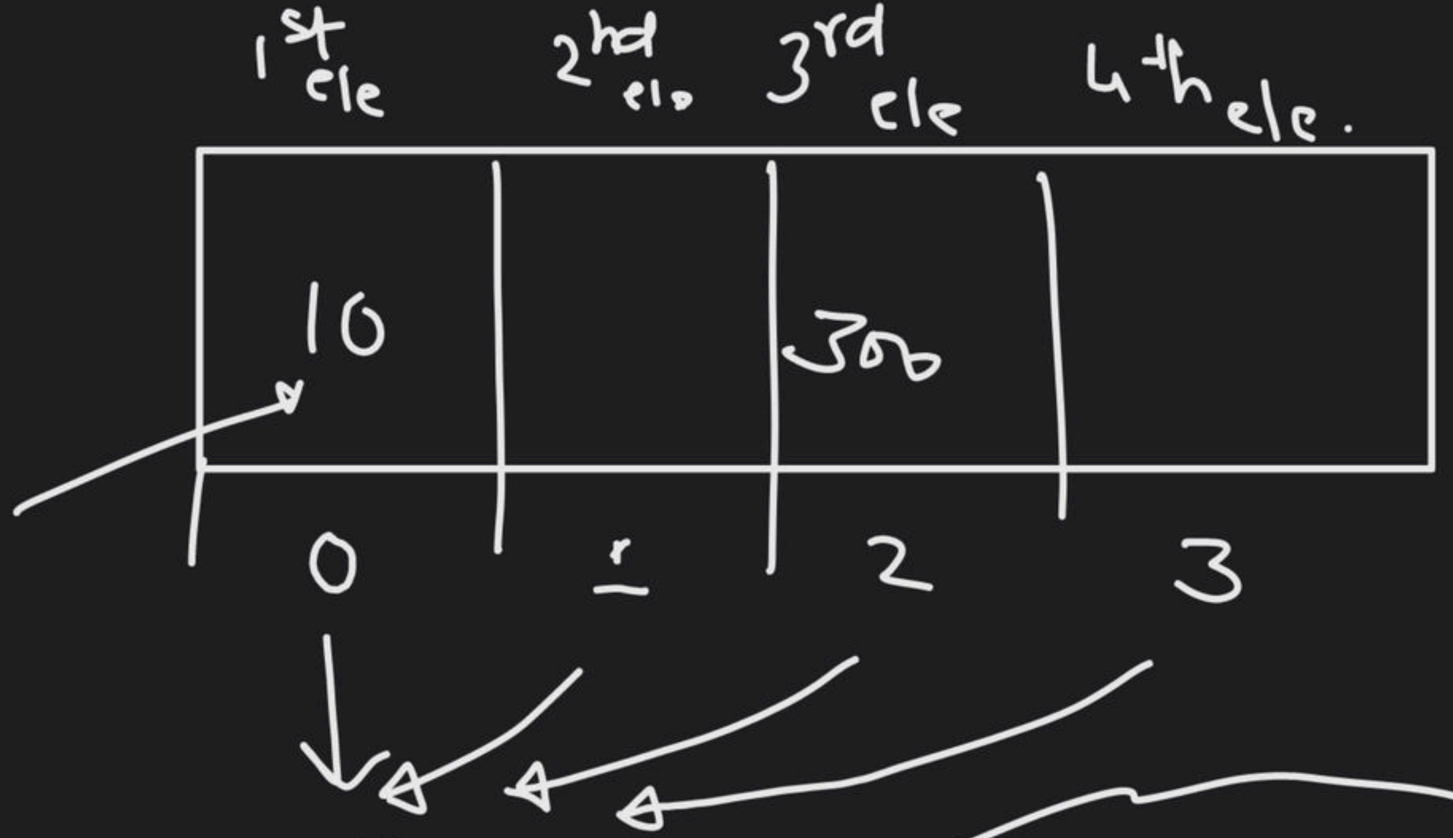
All 3 elements are represented by same name/entity.


Sec-A
A
60
Students
1, 2, 3, ...
Sec-B
B
60
Students
1, 2, 3, ...
unique identification number ⇒ Roll No

unacademy

```
int a[4];

a[0] = 10;

a[2] = 300;
```

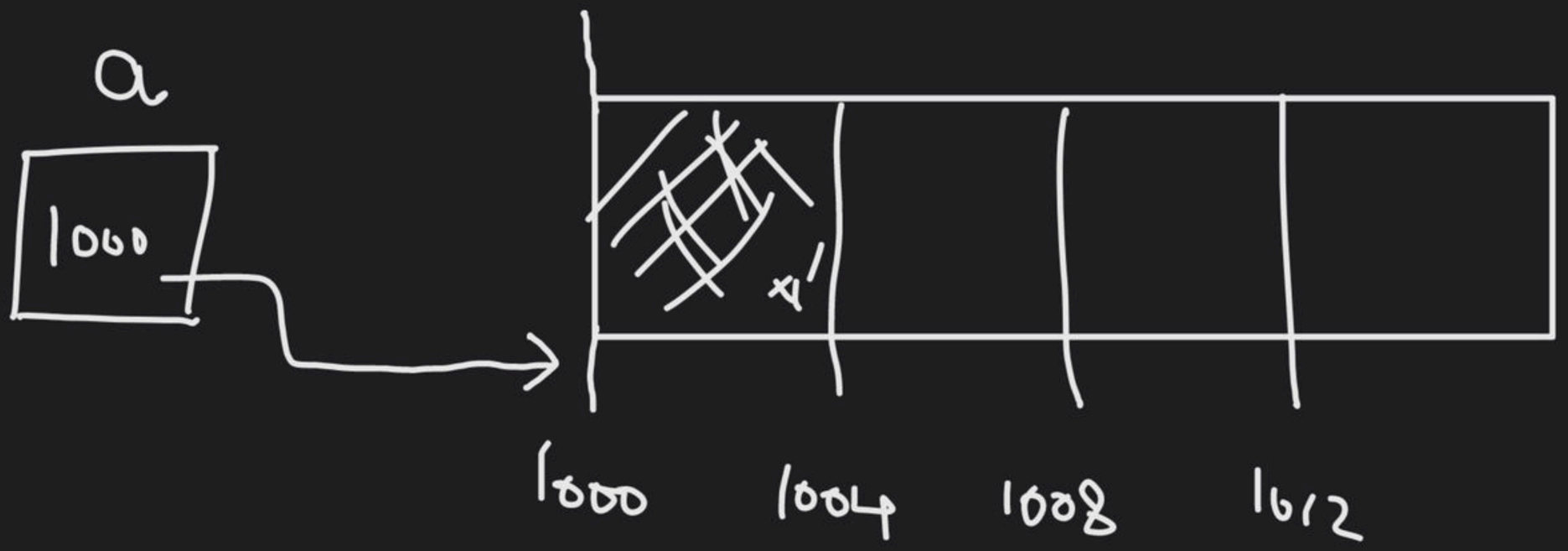
3rd element



(unique id. no.)

In C prog. index always starts from 0

int a[4];

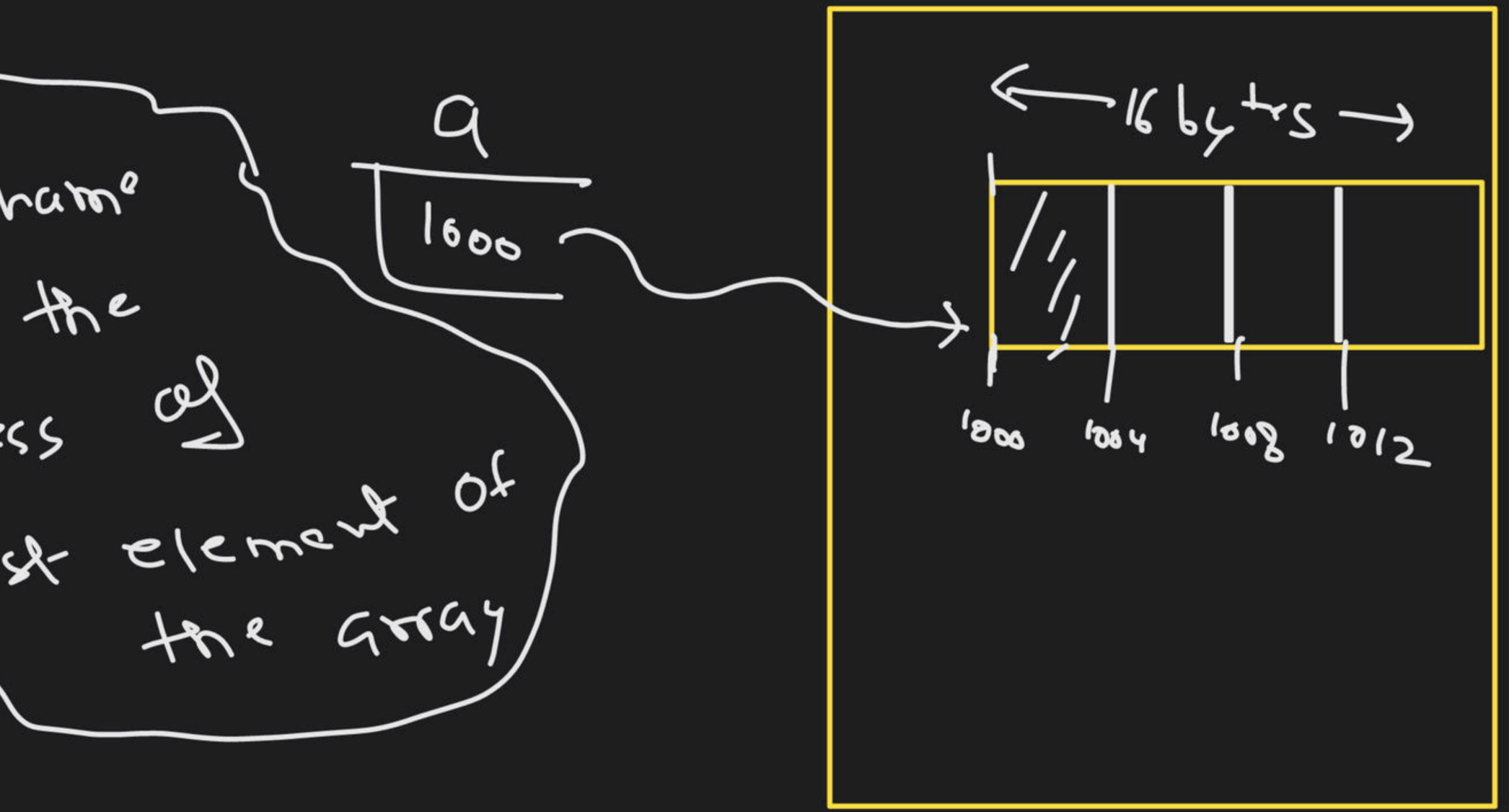


Array name ⇒ address of first element of array

constant add

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

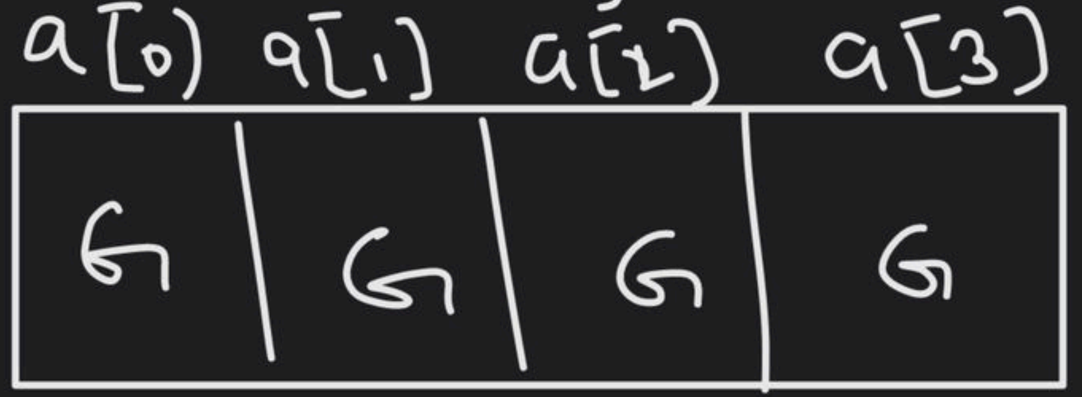
array-name holds the address of first element of the array



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

Garbage

```
void main() {
    int a[4];
    printf("%d", a[0]);
}
```

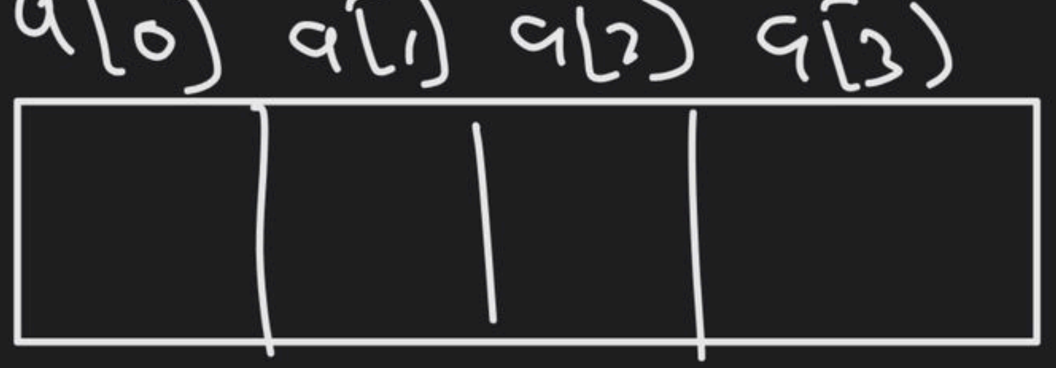


int a, b, c, d;

4 variables of type int

int a[4];

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



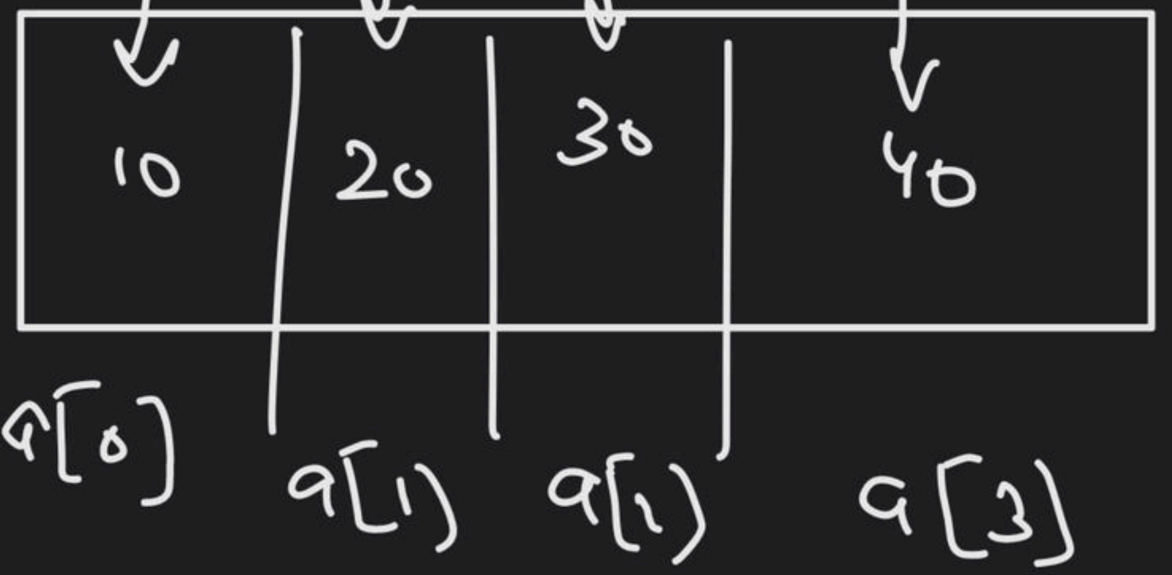
unacademy

```
int a;
```

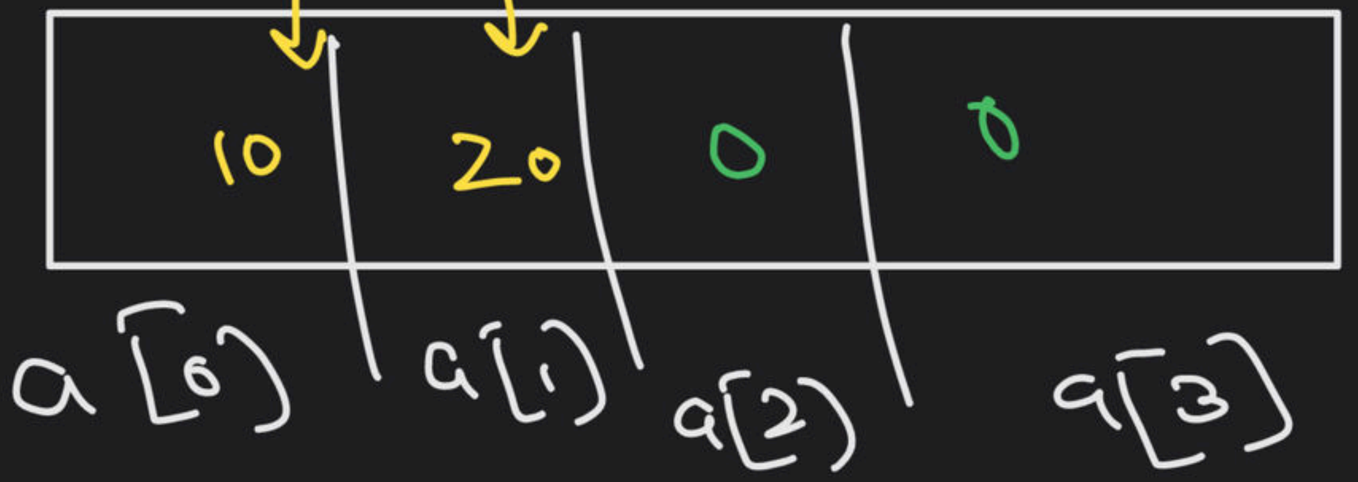
```
int a=1;
```

Initialization

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



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



```
int a[];
```

Invalid?.

group size?.

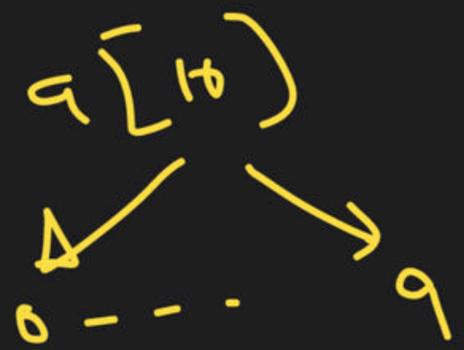
Ud ke lact Pdegi

① `int a[];` Error

② `int a[] = {10,20,30};`

array-size → 3

③ `int a[10]`



unacademy

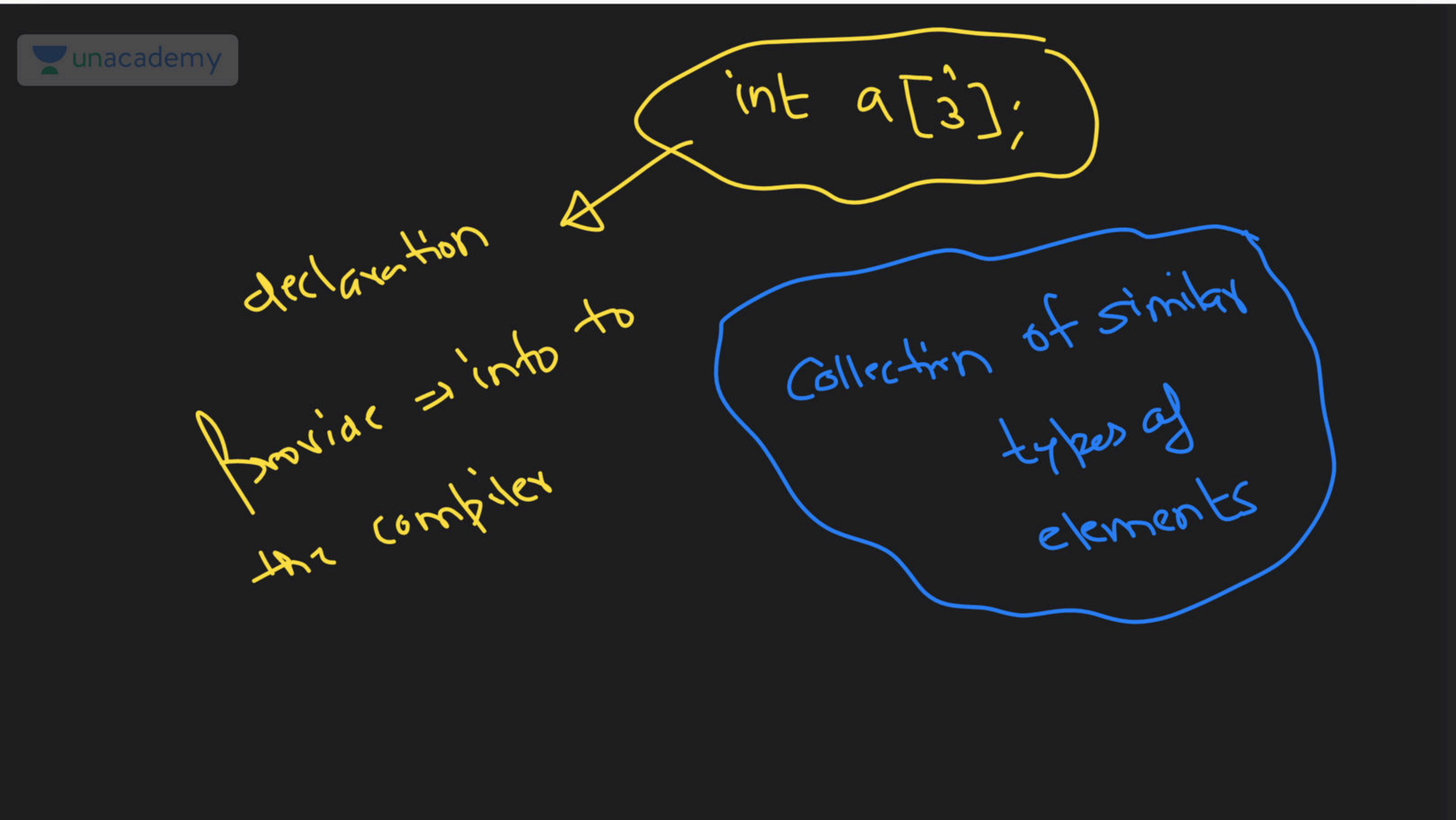
SIZE

```
int a[2+2];

int a[2*3];

int a[4*sizeof(int)];
```

```
#define SIZE 10
void main(){

int a[SIZE];
   ==
}
```

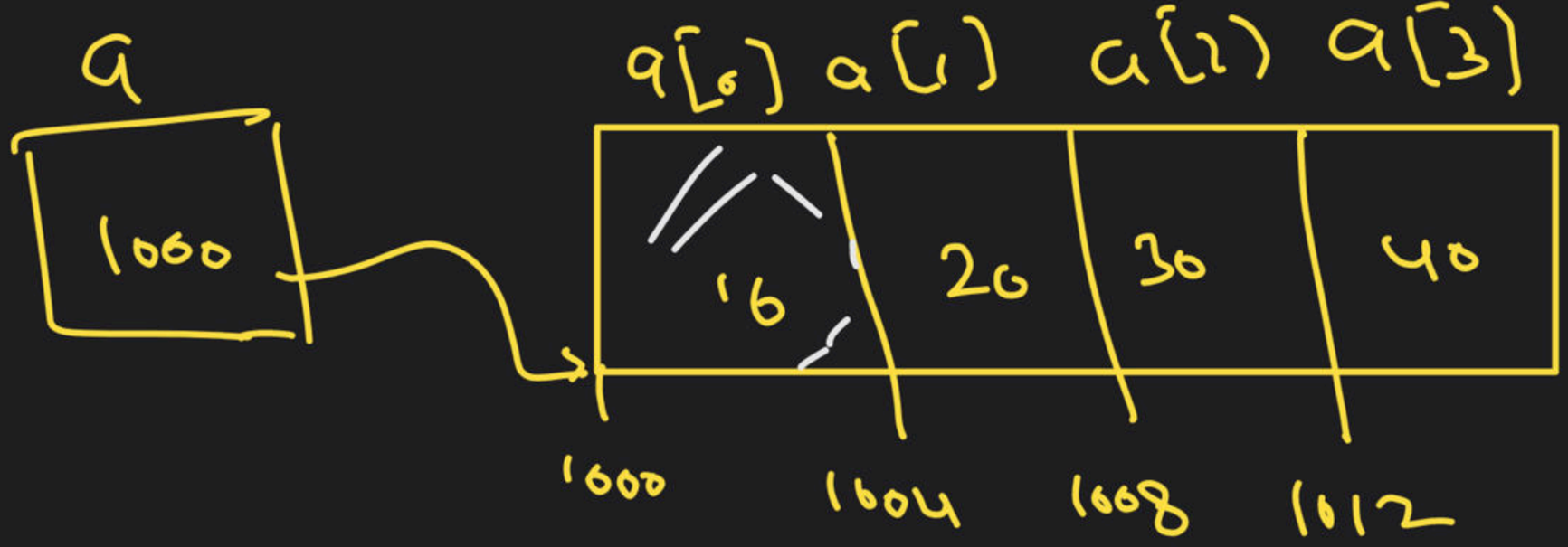
Logically

unsigned int >0


int a[3];
declaration
Provide info to the compiler
Collection of similar types of elements

unacademy

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



Array_name ⇒ constant.

unacademy

Array-name =  ; Invalid Error

Constant

Lvalue X

unacademy

$2 = 10$; Lvalue = Rvalue (Assignment)

must be some variable.

- constant
- Exp.
- variable

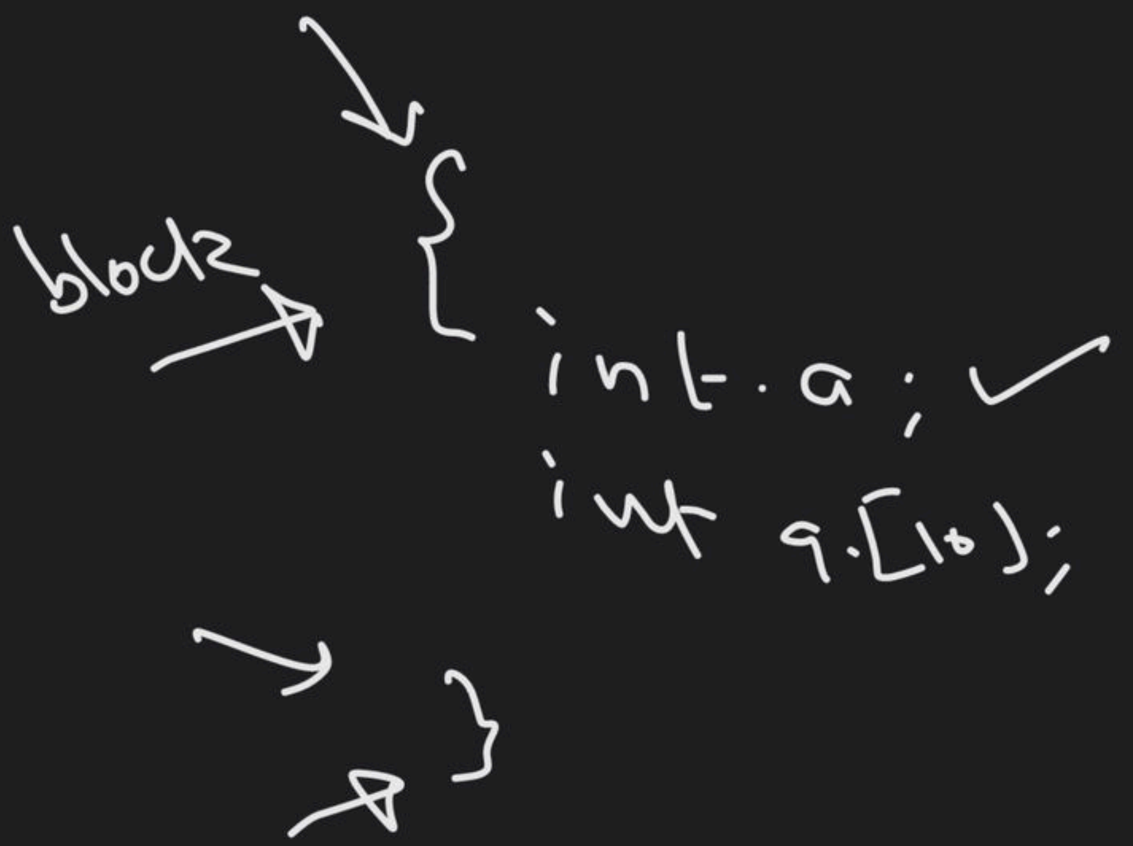
1. Array-name can not be Lvalue of any assignment statement.

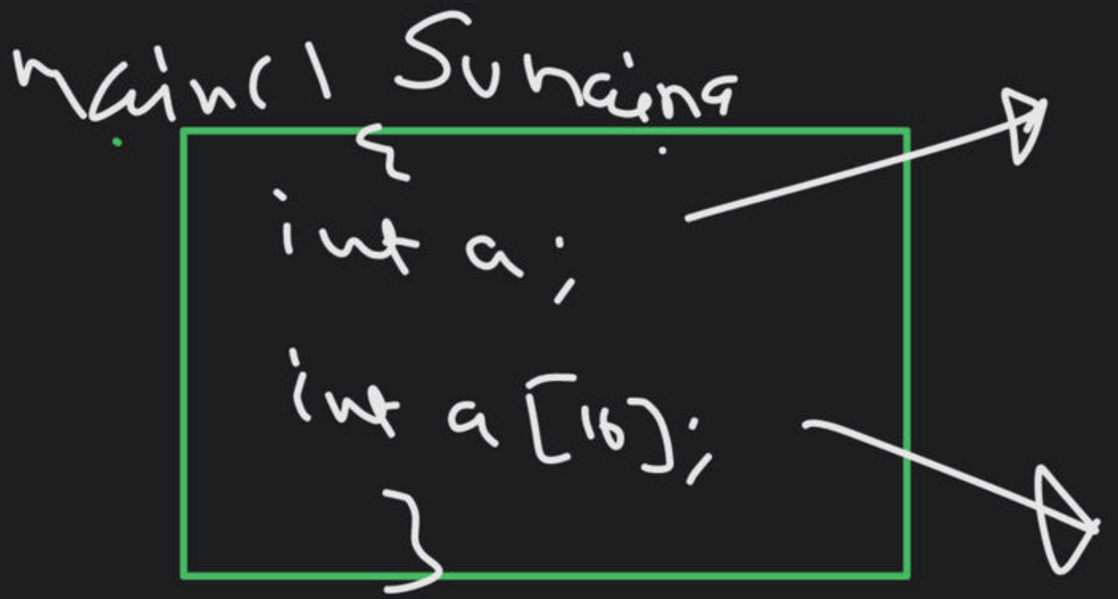
```
void main(){
int a[4]={10,20,30};
printf("%d",a[0]);
}
```

10

```
void main(){
int a[4];
a={10,20,30};
pf("%d",a[0]);
}
```

Can't be Lvalue





(2) Array-name ⇒ Constant.

2++ ; → Invalid

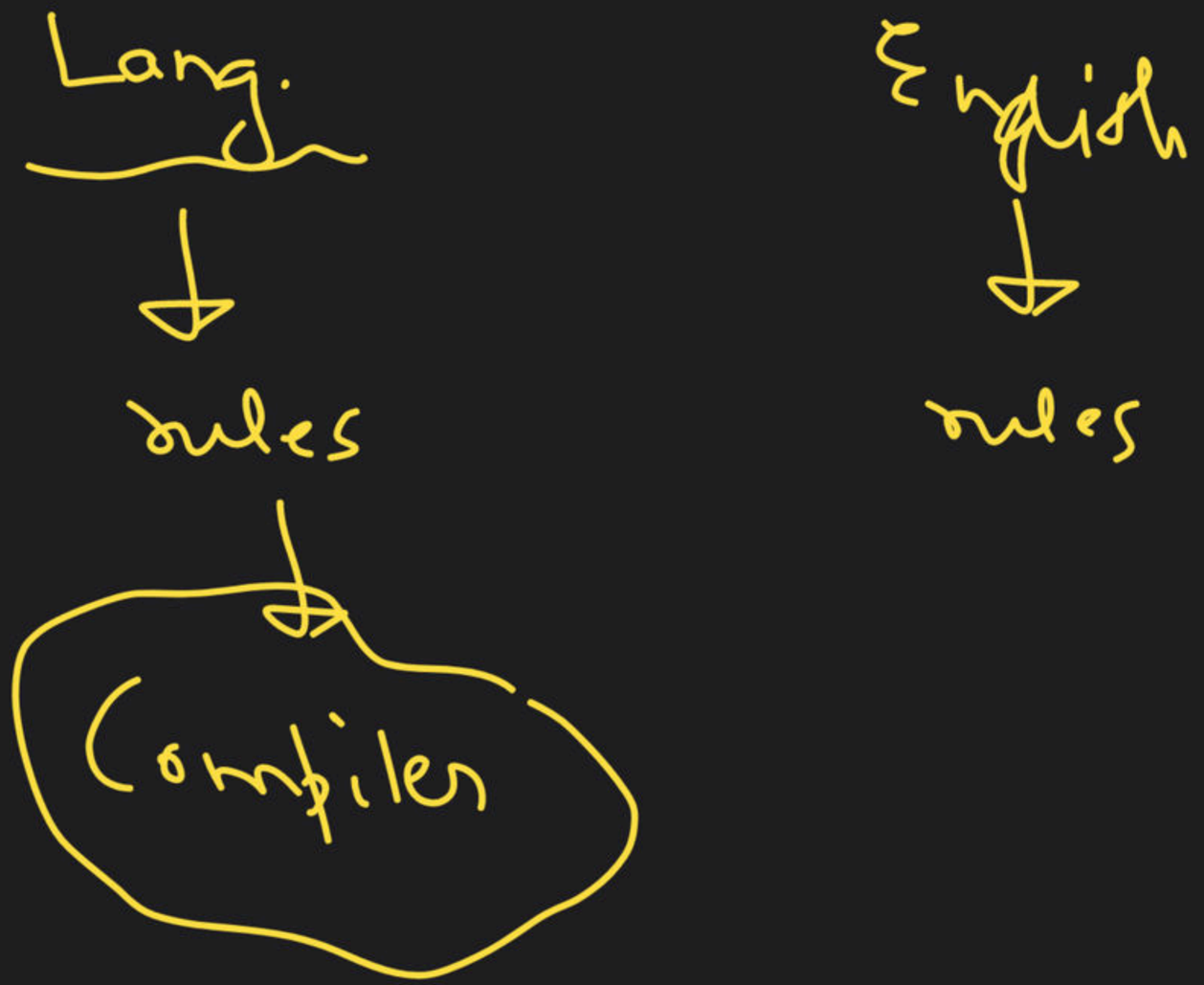
Array-name ++
++ Array-name
Array-name --
-- Array-name
} Invalid.

1) Collection of homo. types of elements.
2) Elements are stored seq. one after another.
3) Purpose ⇒ Relative addressing.
4) Array-name ⇒ constant.
5) Array-name ⇒ Addr. of its first element.
6) Array-name = 0; Invalid

Array-name can not be Lvalue for an assignment statement.

7) ++Array-name
Array-name++
--Array-name
Array-name--
} Invalid.

8) int A[]; Invalid.



Array concept.

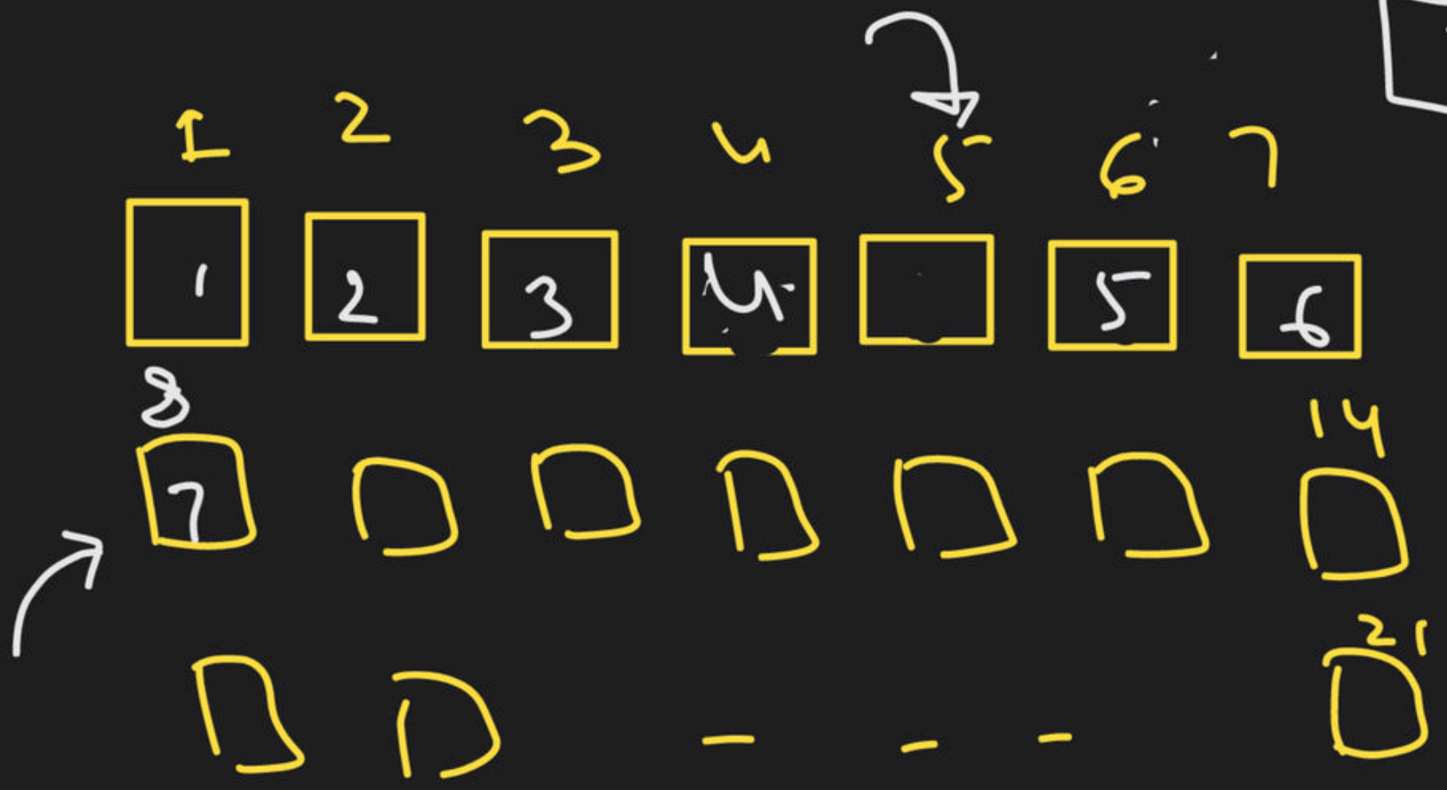
```
int a[10] = {10,20};
int a[4] = {10,20,30,40};
```

Collection of many variables

Collec. of values

Relative add.

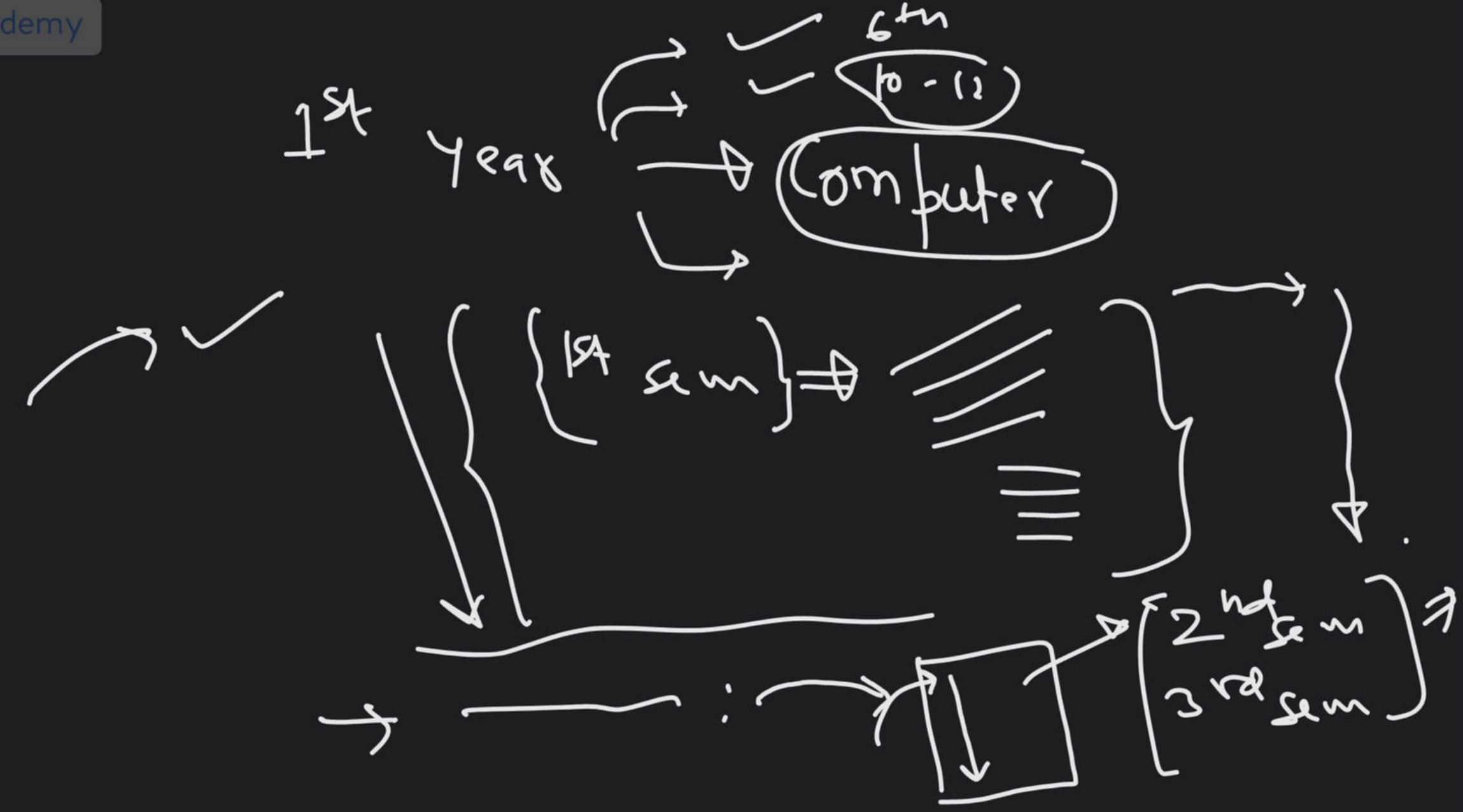
- cont.
- same type



```
int a = 1;
int a = {1};
```

valid

```
int (a) = 1;
```



```
int a = 1;
int a[] = {1};
```

22 :

variable Aditya

variable / constant

a+1

2++

Array name ⇒ Constant ⇒ Constant ++

Array-name ++

a[2]++ ;

unacademy

THANK YOU!

Here's to a cracking journey ahead!