

WHY WE NEED ARRAY IN PROGRAMMING ?

WHAT IS ARRAY?

TYPES OF ARRAY

HOW TO DECLARAEE AN ARRAY IN C

INITIALIZATION OF AN ARRAY

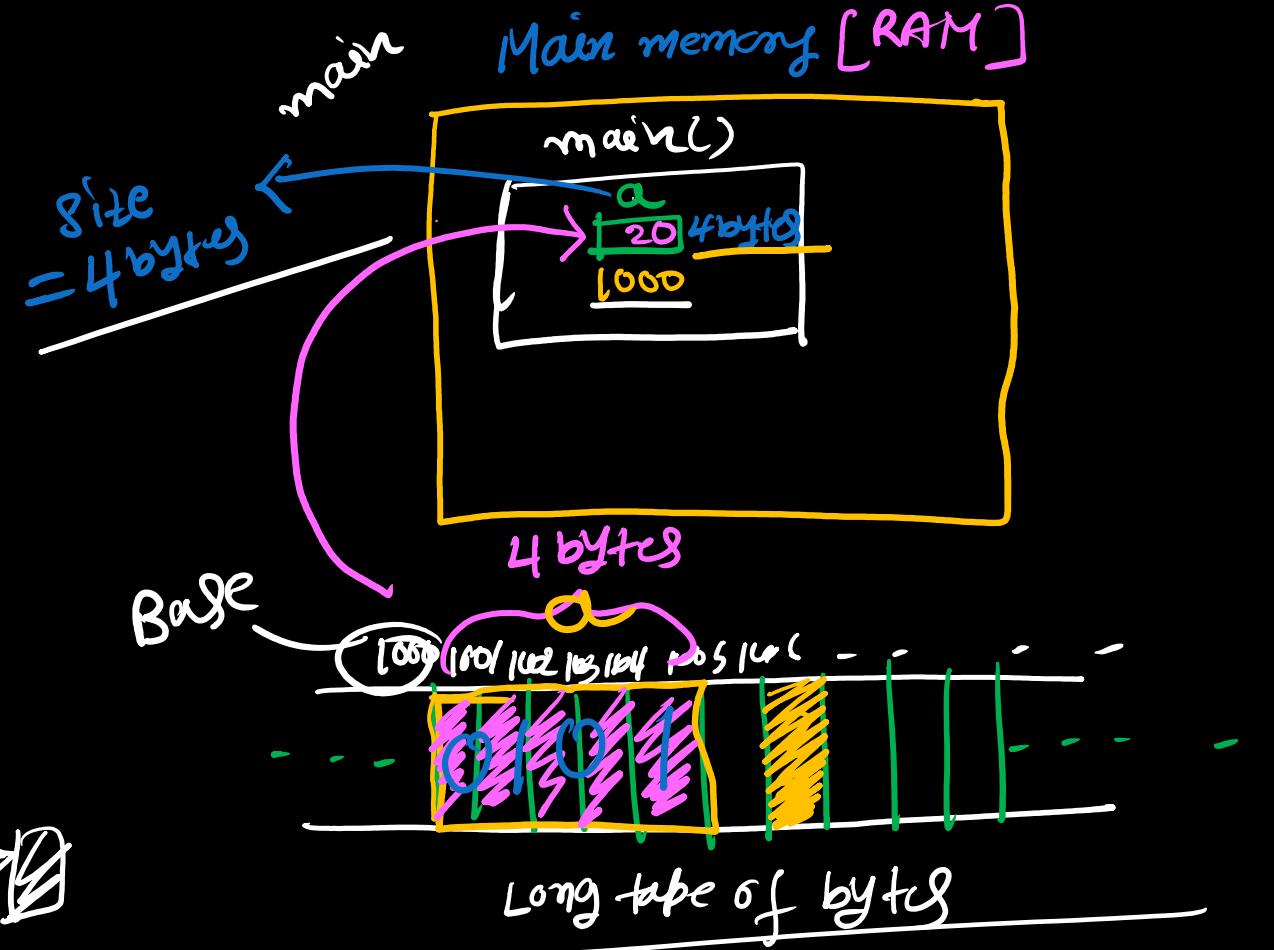
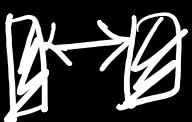
main()

```
{  
    int a = 5, 10, 20;  
    → a = 10;  
    → a = 20;  
}
```

why we need array

a var location/
address

address assume = Base address
= 100, 200, 1000,
1024



1D array

in my college i have 60 Students Now I want to store the roll-no of 60 Students

~~a1 = 106
a2 = 101
a3 = 102
a4 = 103
. .
;
a60 = 168~~

Array

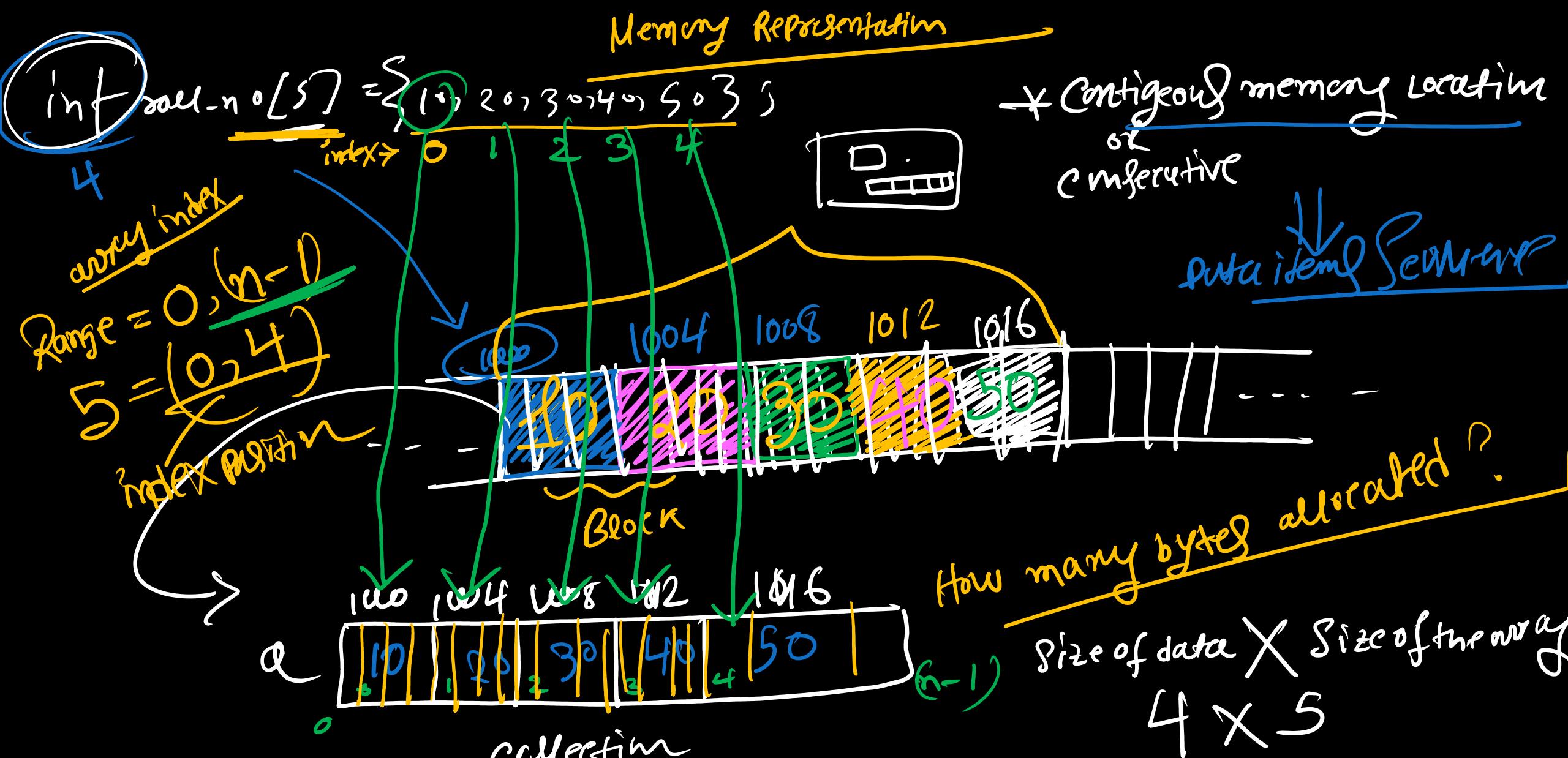
datatype nameofarray [size]

int roll-no [5] = {16, 20, 36, 46, 59};

↓
Declaration - initialization

Value = Data item or element

int roll-no [60] memory = 240



* Contiguous memory location

or
consecutive

↓
data item sequence

$$\begin{aligned}
 & 4 \times 5 \\
 & = 20
 \end{aligned}$$

what's array?

- ↳ ① collection of more than one data item
- ↳ ② all the data items must be be same type

{ 1, 2, 3, 4, 5 }	✓	float
{ 1, -2, 10, 0, -5 }	✓	
<hr style="border: 1px solid yellow; margin-bottom: 10px;"/> <u>{ 10, 20, 0, -5, 10.5, -5 }</u>	X	assume int

int a[5];

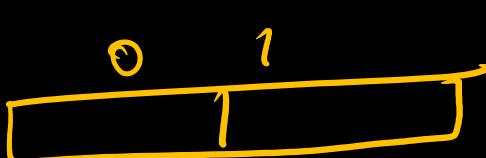
char a[5] = { 'a', 'b' }

float a[5] = { 10.5, 2.5 }

① $\text{int } a[-];$ → error ✓
no size mention?

② $\text{int } a[-5];$ X → error
any positive constant value

③ $\text{int } a[4/2];$ ✓



~~$a[4]$~~
 ~~$a[10]$~~
 ~~$a[11]$~~
 ~~$a[10/2]$~~
 ~~$a[11/2]$~~

④ $\text{int } a[6/3*2];$ ✓

⑤ $\text{int } a[5/10/2];$ ✓

~~$a[11/10/5]$~~

~~$b = 11/2$~~

⑥ $\text{int } b;$
 $\text{int } a[b = 11/2];$ X

$\text{int } a[3-2];$ ✓
 $\text{int } a[11/2];$ ✓



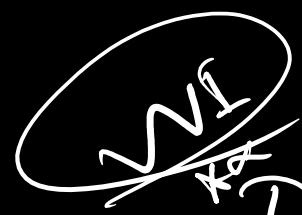
Data type

Primitive

[int, float, char, bool,
double]

Non-Primitive

[array, Structure, union]



Data structure

Linear dS

Memory sequence

- ↳ array 
- ↳ Linked List 
- ↳ Stack -
- ↳ Queue -
- ↳ Deque -
- ↳ Circular Queue -

Non Linear

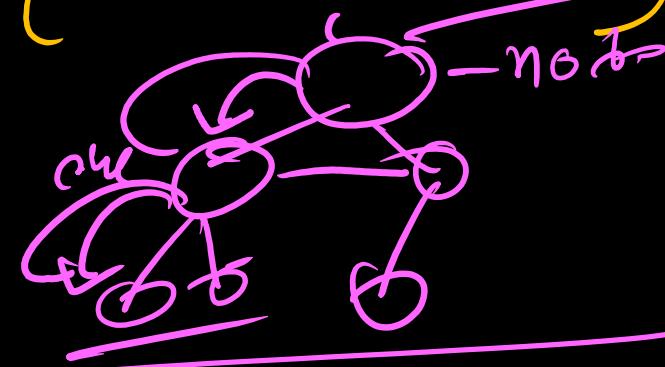
No sequence

- ↳ Tree
- ↳ Graph
- ↳ Dynamic programming
- ↳ Programming

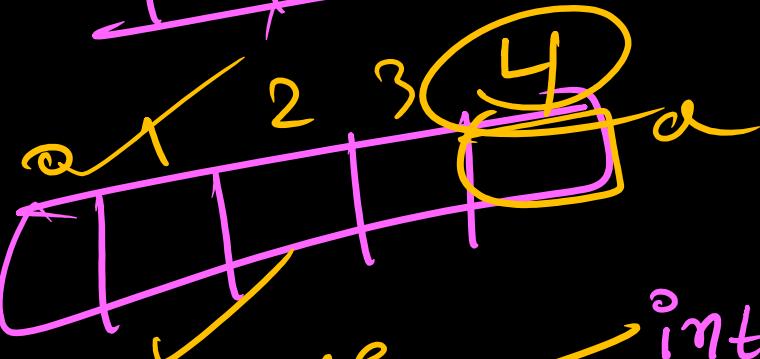
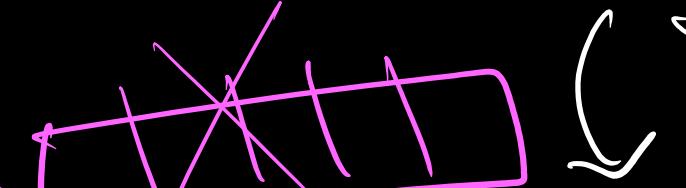
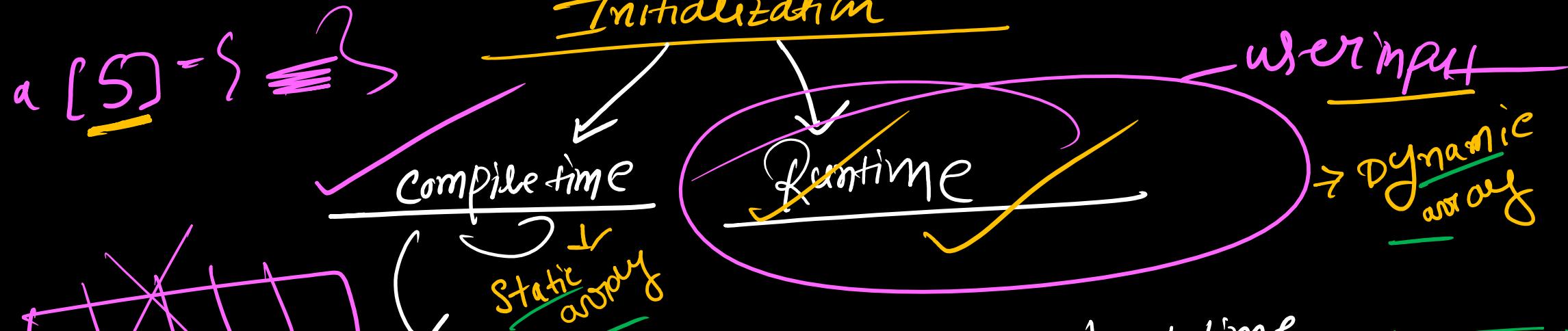
Recursion *

* array non primitive datatype

{ 'a', 'b', 'c', 'd' }



vector &



advantage

- ① code optimization
- ② index find

$\text{ps}(a[4])$

when you declare the array that time
you initialize the array -

int a[5] = { '1', '0', '6', '-1', '2', '5' };

size fixed
disadvantage

① int a[5] = {10, -1, 11, 15, 2} ✓

② int a[] = {10, 2, 3, -5, 6, 20} ✓
no size

* compiler automatically calculate the bytes

$$4 \times 6 = 24 \text{ bytes}$$

0	1	2	3	4	5
10	-2	3	-5	6	20



size fixed

5 < item ✓

③ int a[5] = {0, 1, 2} ✓

0	1	2	3	4
0	1	-1	0	0

X

④ int a[5];

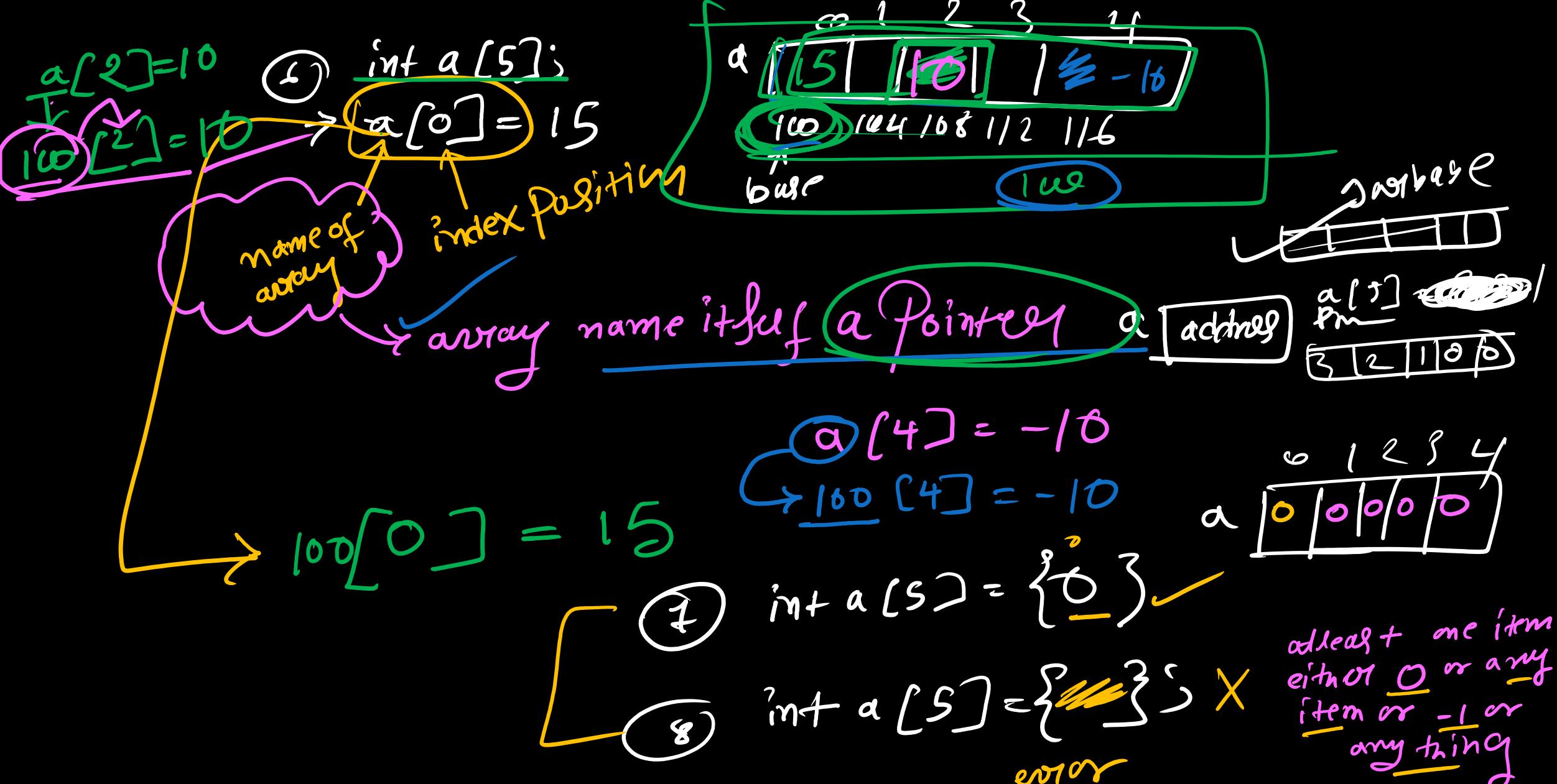
0	1	2	3	4
#	#	#	#	#

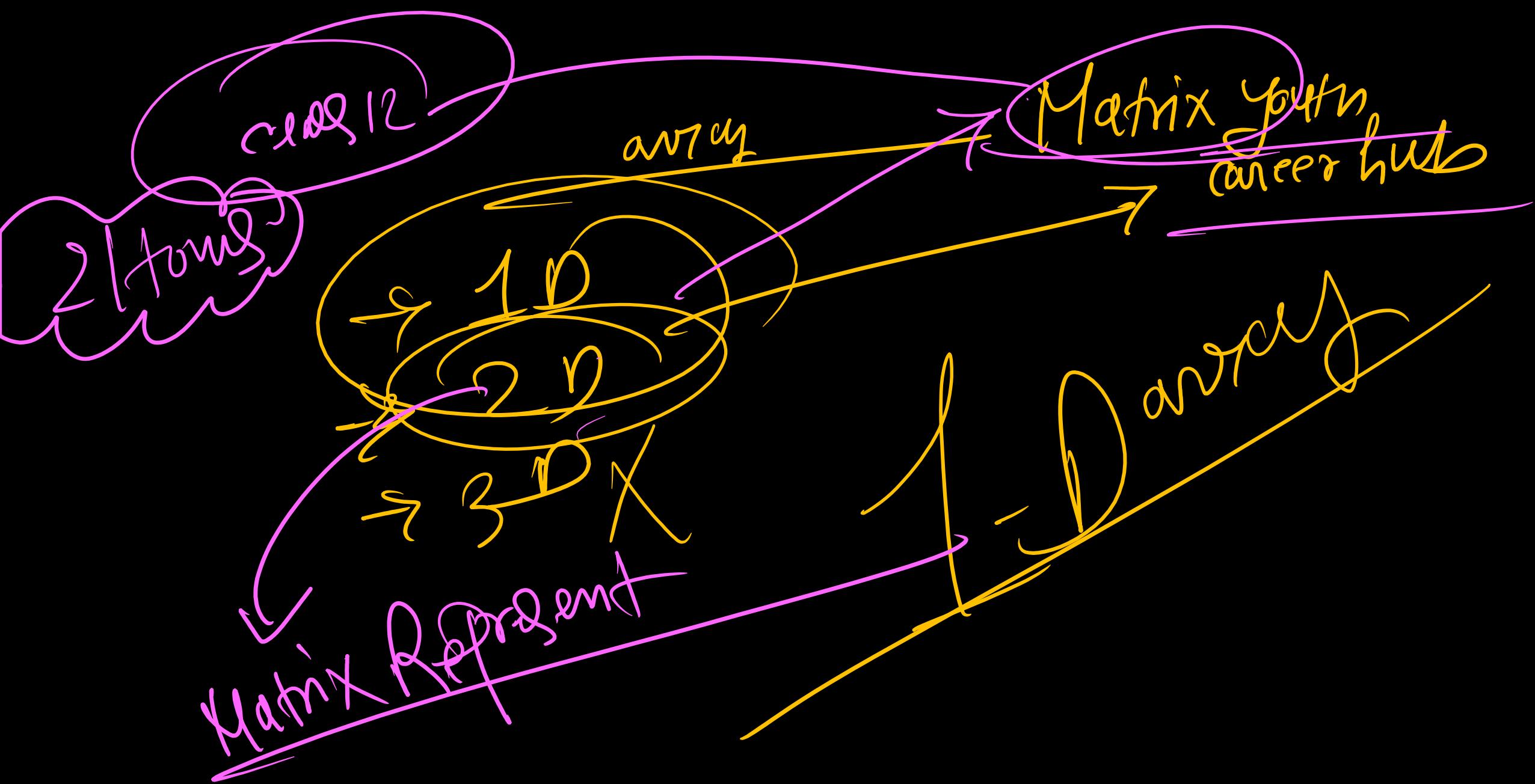
garbage value

⑤ int a[5] = {1, 2, 3, 4, 10};

0	1	2	3	4
1	2	3	4	10

BS(a) > garbage



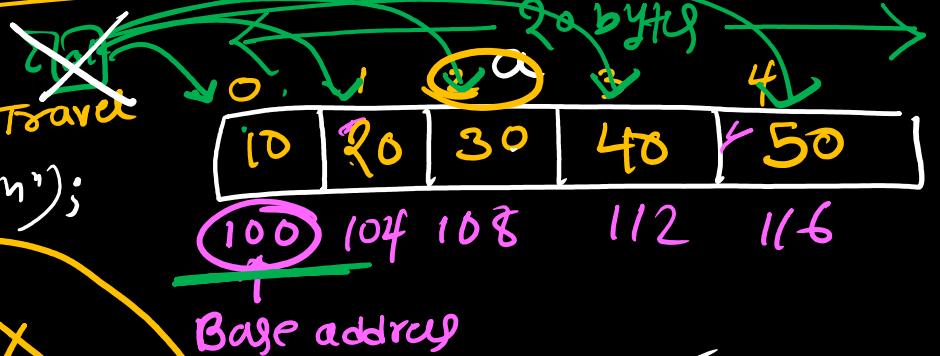


$a[]$

Runtime initialization

✓ `int a[5], i;`
~~printf("Enter the element:\n");~~

I/P X
~~scanf("%d", &a[i]);~~
~~for (i=0; i<5; i++)~~
~~{~~
~~i = 0 ↓ ↓ ↓ ↓ ↓~~
~~scanf ("%d", &a[i]);~~
~~100 [0]~~
~~100 [1]~~
~~105 [2]~~
~~110 [3]~~
~~115 [4]~~
~~}~~
~~for (i=0; i<5; i++)~~
~~{~~
~~i = 0 ↓ ↓~~
~~printf ("%d", a[i]);~~
~~100 [0]~~
~~100 [1]~~
~~105 [2]~~
~~110 [3]~~
~~115 [4]~~
~~}~~
~~return 0;~~



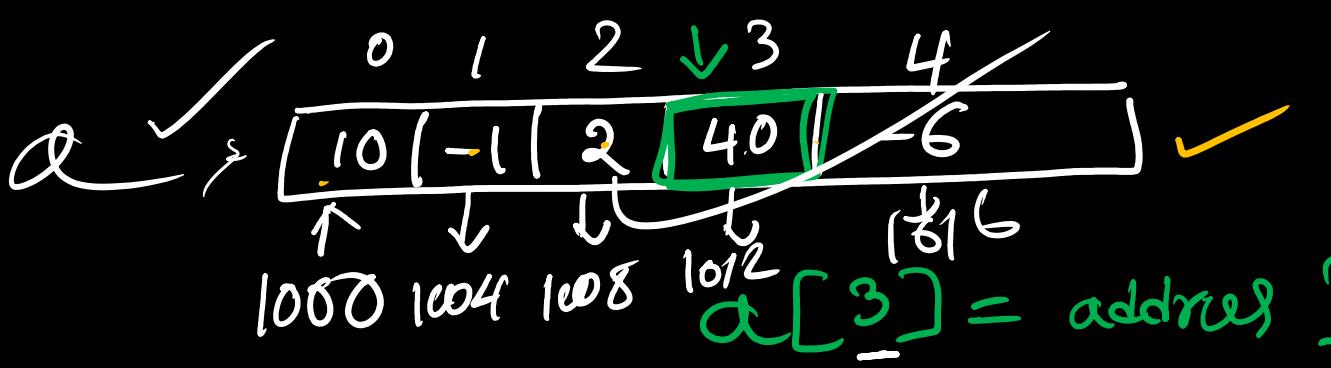
Traversal algorithm
 COSA
 index
 i

{
~~i~~ 100 [0] 105 [1] 110 [2] 115 [3] 120 [4] 125 [5]

Console window

Enter the element
~~-10~~
~~-20~~
~~-30~~
~~-40~~
~~-50~~
 I/P stop

10
20
30
40
50



100

$$\begin{aligned}
 &= B + i \times S \\
 &= 1000 + (3 \times 4) \\
 &= 1000 + 12 \\
 &= 1012
 \end{aligned}$$

$B = \text{base address}$
 $i = \text{index}$
 $S = \text{datatype size}$

* Read 5 Students' Marks → calculate Sum and Avg

array program [1-D array] (5 marks)

3
x

a1

2 marks				
1	2	3	4	5
0	1	2	3	4

a2

--	--	--	--	--

$$\text{avg} = 15 | S = 3$$

```
int marks[5]; marks(?)  
int sum=0, avg=0;  
for (i=0; i<5; i++)  
{  
    scanf ("%d", &marks[i]);  
}
```

sum [] 0

avg. [] 3

```
for (i=0; i<5; i++)  
{  
    sum = sum + marks[i];  
}
```

i [0] + 2 3 4 5

$$\text{sum} = 0 + 1 = 1$$

$$\text{sum} = 1 + 2 = 3$$

$$\text{sum} = 3 + 3 = 6$$

$$\text{sum} = 6 + 4 = 10$$

$$\text{sum} = 10 + 5 = 15$$

avg = sum / 5;

pf (avg) 3
pf (sum) 15

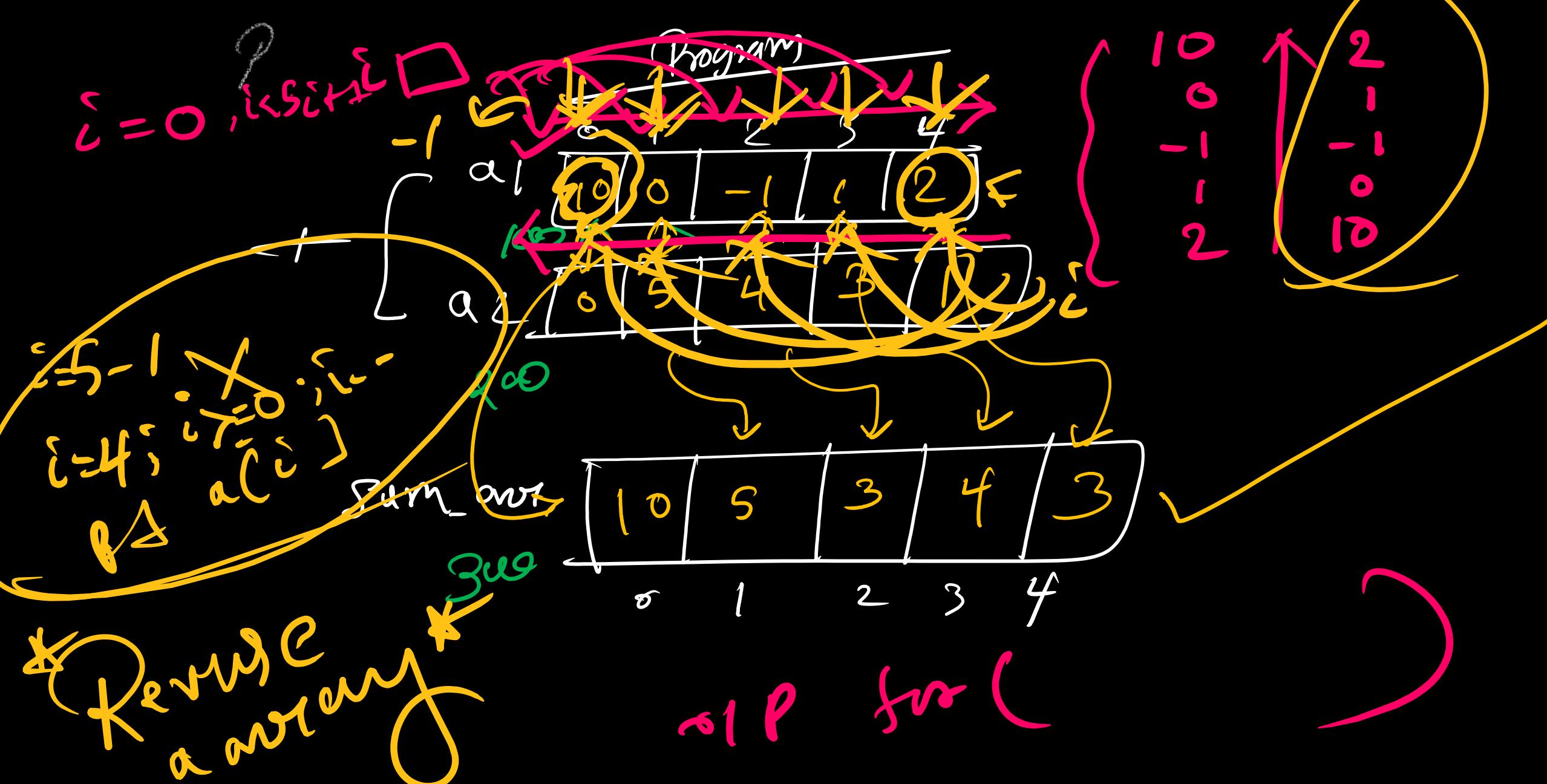
100 [0]
100 [1]
100 [2]
100 [3]
100 [4]

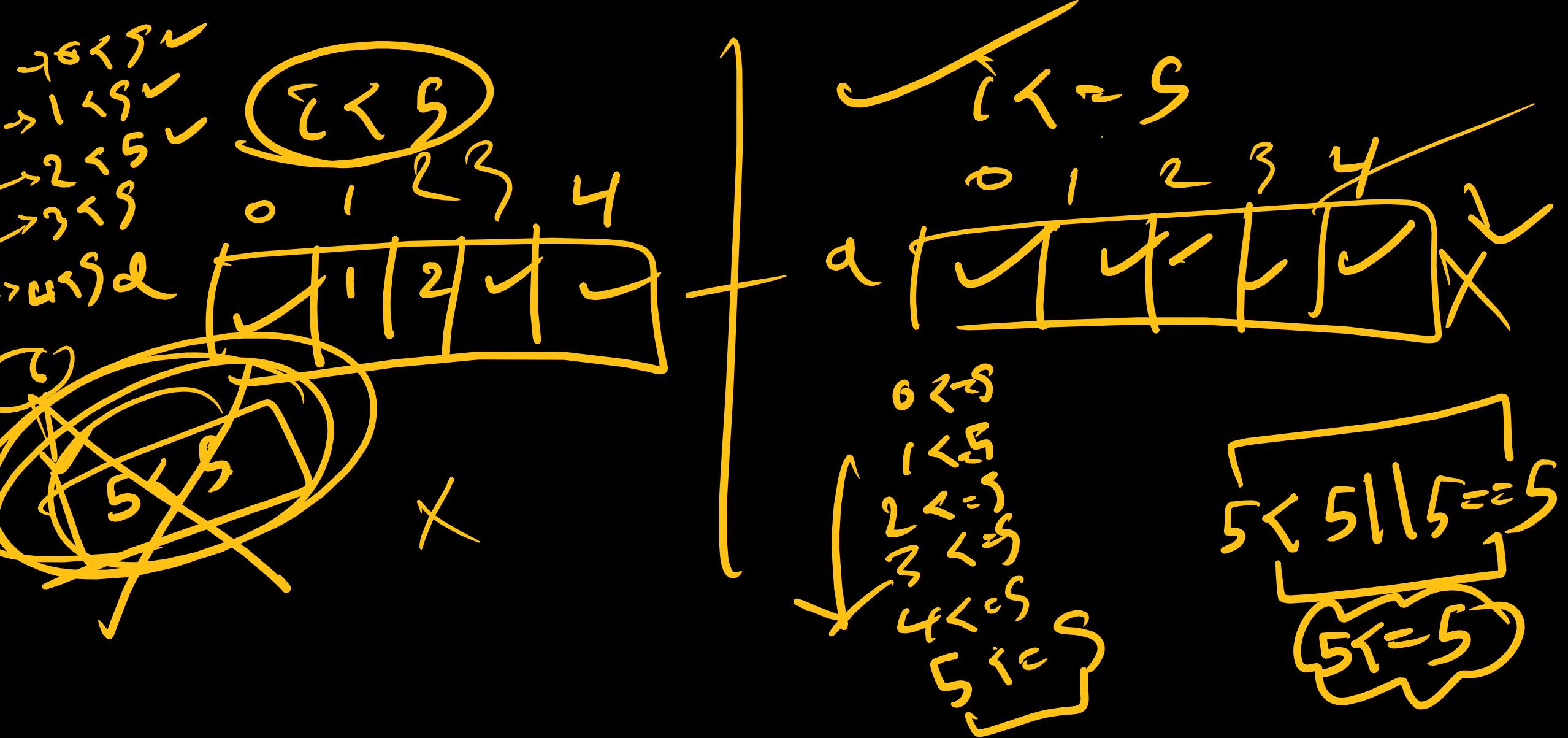
array [10]



5 odd

5 even





20-22

30

Coupled F

1 Hairs

Inhorn al

cat

MCQ

