

# C Language LIVE Community Classes

## Arrays

Day-16

By

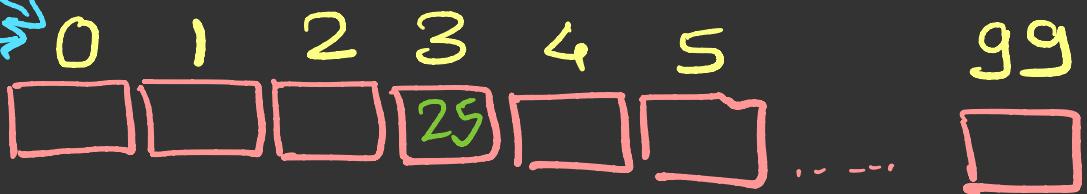
Saurabh Shukla  
(MySirG)

## Introduction to Arrays

- Array is a linear collection of similar elements.
- Array is also known as Subscript variable

Write a program to calculate average of  
100 numbers. index → subscript value

int a[100];



a[3] = 25;  
expression

$m_1 m_2 m_3 \dots m_{100} \leftarrow$  subscript

$m[1] m[2]$

i=0 to 99

[ ] subscript operator

scanf("%d", &a[i]), 3 operands

a[i] = 25;      x + y    3 + 4

## Array Declaration Rules

- ① int a[]; Error  
Can't be empty
- ② int a[5];
  - Natural number
  - Total number of variables in array
  - Not an index
- ③ int a[5];

0	1	2	3	4

local array when not initialized contains garbage values.

whatever is the size of an array it always consumes memory in a sequential fashion.

④ You can initialize array during declaration

```
int a[5] = {10, 50, 30, 70, 20};  
          0   1   2   3   4  


|    |    |    |    |    |
|----|----|----|----|----|
| 10 | 50 | 30 | 70 | 20 |
|----|----|----|----|----|


```

⑤ You cannot initialize an array during declaration more than its size

```
int a[5] = {10, 50, 30, 70, 20, 80, 40};
```

Error

⑥ You can initialize an array during declaration with lesser values than the size of an array.

int a[5] = {10, 50};

0 1 2 3 4

10	50	0	0	0
----	----	---	---	---

Remaining variables in array will contain 0.  
and not garbage value.

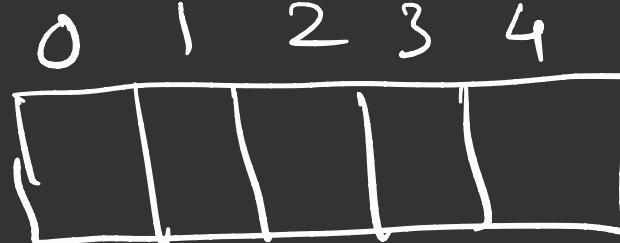
⑦ During declaration you can leave [ ] empty  
only when you initialize array at the same time.

int a[] = {10, 50, 30, 80, 20};

## Bound Checking

int a[5] = {10, 20, 50, 90, 30, 60, 70}; Error

int i, a[5];



for (i=0; i<=9; i++)  
scanf("%d", &a[i]);

# Sorting

- Arranging elements in some logical order is known as sorting.
- By default, for numbers sorting means arranging elements in ascending order.

example

0 1 2 3 4 5 6 7

given array →

20 50 90 60 70 80 30 10

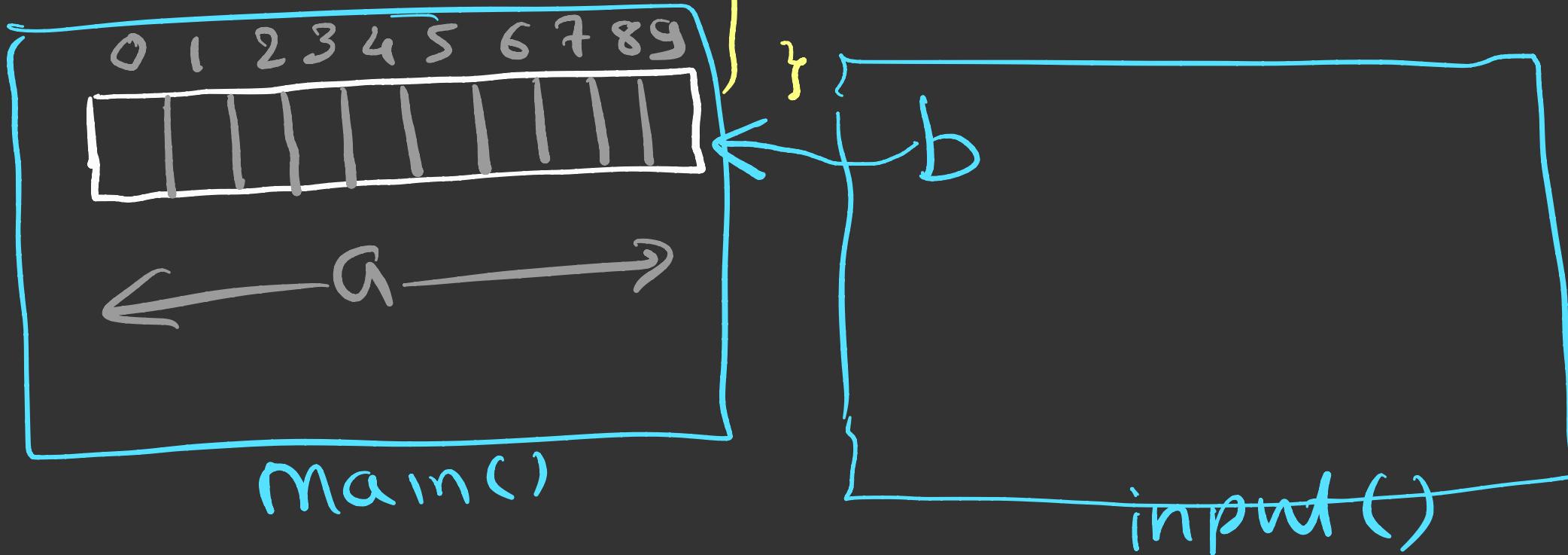
Sorted array → 10 20 30 50 60 70 80 90

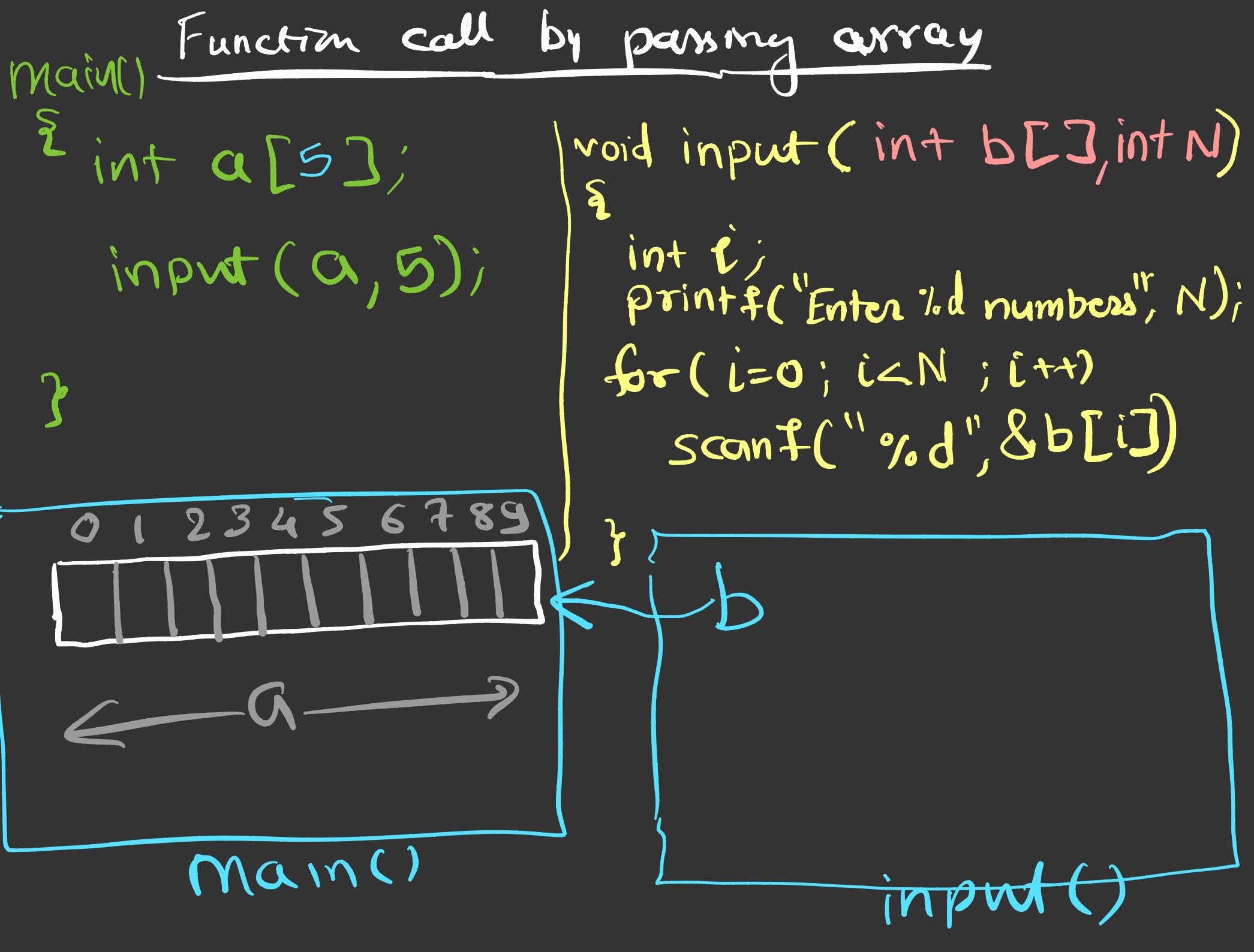
Function call by passing array

```
main()
{
    int a[10];
    input(a);
}
```

```
void input( int b[] )
```

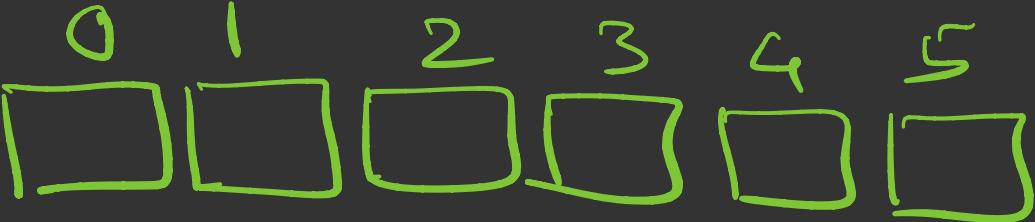
```
{  
    int i;  
    printf("Enter 10 numbers");  
    for( i=0 ; i<=9 ; i++ )  
        scanf("%d", &b[i]);
```





## Two dimensional Arrays

int a[6];



one dimensional Array

int b[2][3];



Two dimensional

int c[4][5][3];

Three dimensional



① age 23 int age, date, f, m, marks . . .

④ Date 3

# ③ Family 4

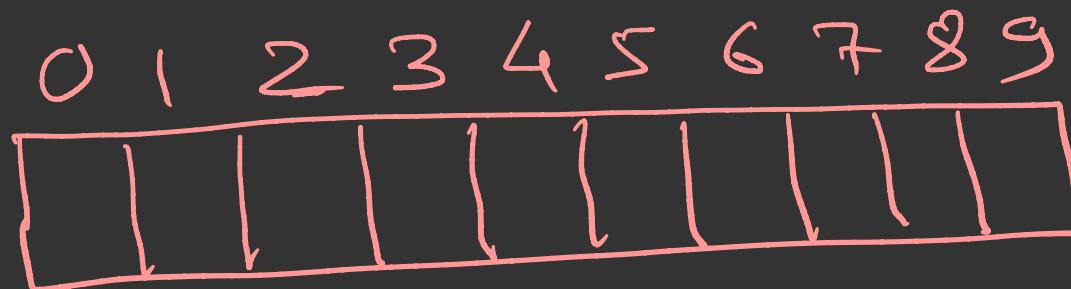
# Movies

(5) Marks 25

0	1	2	3	4	5	6	7	8	9
25	37	42	83	69	21	40	50	10	5

```
int a[10];
```

int a[10];



int a[50];



0

1

2

3

4



int a[5][10];

# Multi-Dimensional Arrays

int a[3][5][4];

0

0	1	2	3	4
0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3
0	0	0	0	0

1

0	1	2	3	4
0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3
0	0	0	0	0

2

0	1	2	3	4
0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3
0	0	0	0	0

```
int a[3][4];
```

```
int i, j;
```

```
for (i=0 ; i<=2 ; i++)
```

```
{
```

```
    for(j=0 ; j<=3 ; j++)
```

```
        scanf("%d", &a[i][j]);
```

```
}
```

i            j  
0            0  
0            1  
0            2  
0            3  
1            0  
1            1  
1            2

1            3  
2            0  
2            1  
2            2  
2            3