

A job ready bootcamp in C++, DSA and IOT

Arrays



Saurabh Shukla (MySirG)

Agenda

- ① Introduction to Arrays
- ② Array declaration rules
- ③ Bound Checking
- ④ Sorting
- ⑤ Function call by passing arrays
- ⑥ Two dimensional arrays
- ⑦ Multi dimensional arrays.

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.

```
int main() index
{   subscript value
    int a[100], i, sum = 0;   a[3] = 30;
    float avg;               [ ] subscript operator
    printf("Enter 100 nos");
    for(i=0; i<=99; i++)
        scanf("%d", &a[i]);
    for(i=0; i<=99; i++)
        sum = sum + a[i];
    avg = sum / 100.0;
    printf("Average is %.f", avg);
    return 0;
}
```

Diagram illustrating the array a:

The diagram shows an array a with indices from 0 to 5 and subscript values 1, 2, 3, 4, 5, 99. Brackets above the indices indicate the range from 0 to 5, and arrows point to the first element at index 0 and the last element at index 5. Below the indices, there are six empty boxes representing memory locations, followed by a dot-dot operator, and another empty box representing the rest of the array.

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.

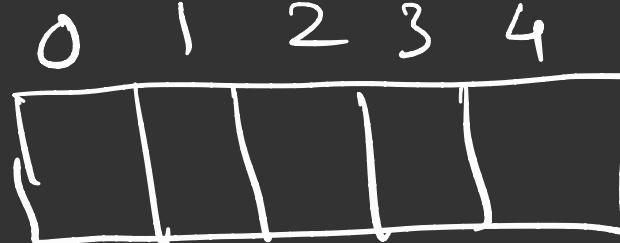
⑦ During declaration you can leave [] empty
only when you initialize array at the same time.

```
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

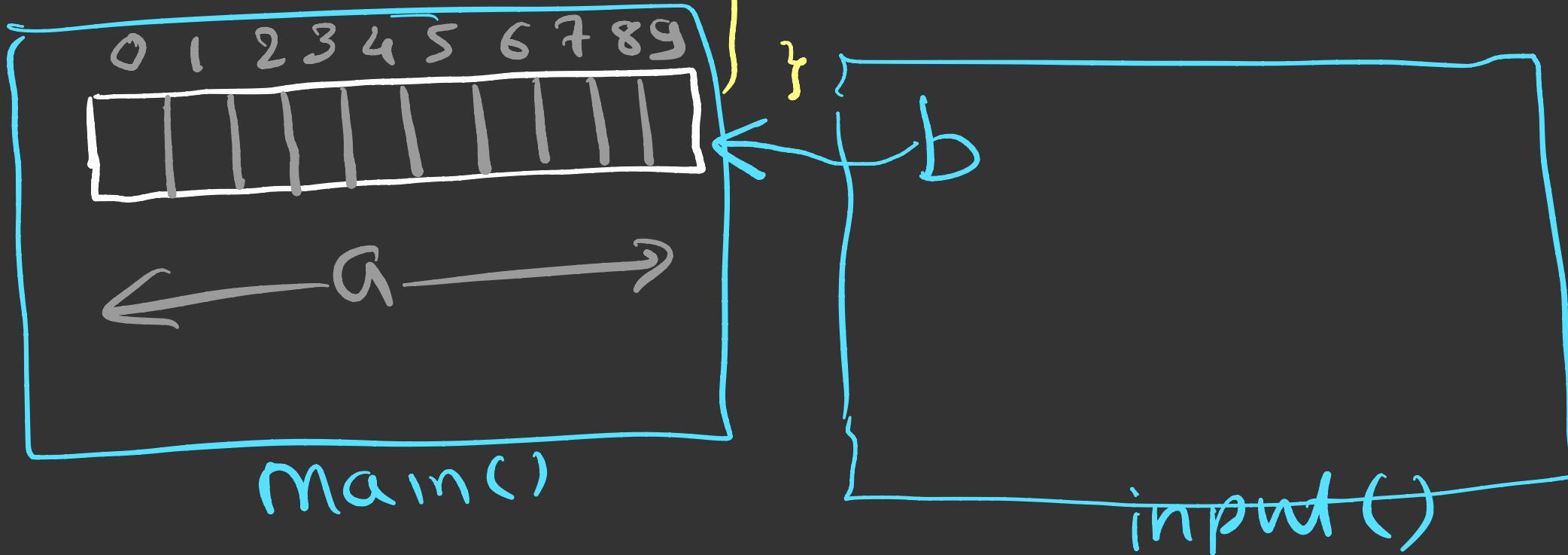
Sorted array → 10 20 30 50 60 70 80 90

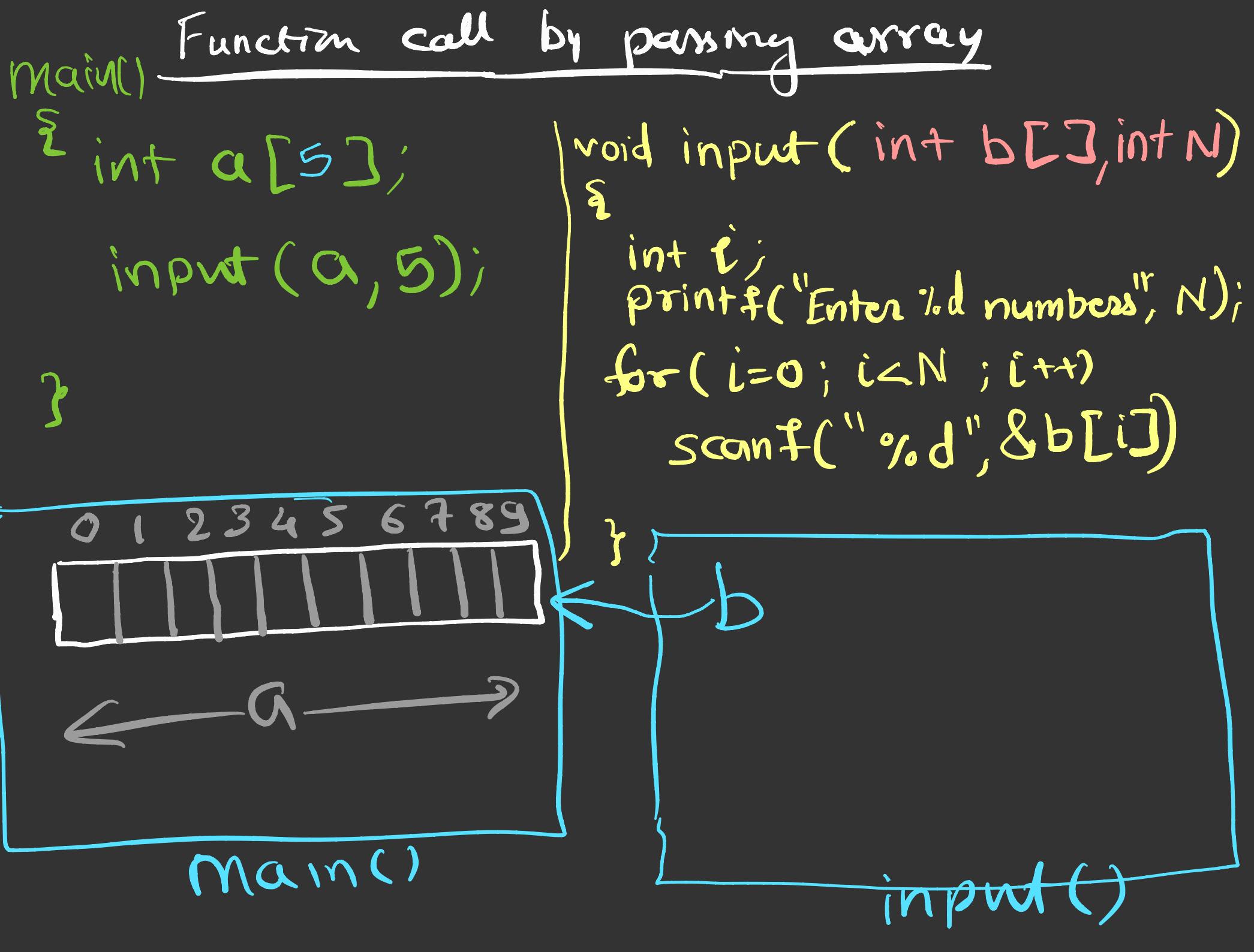
Function call by passing array

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

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

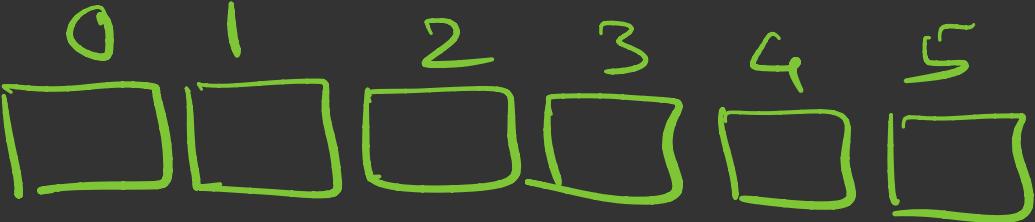
```
{  
    int i;  
    printf("Enter 10 numbers");  
    for( i=0 ; i<=9 ; i++ )  
        scanf("%d", &b[i]);
```





Two dimensional Arrays

`int a[6];`



one dimensional Array

`int b[2][3];`



Two dimensional

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

Three dimensional

int a[50];

int a[5][0];

A hand-drawn number line starting at 0 and ending at 9. The numbers are written above the line, and vertical tick marks below the line indicate integer values from 0 to 9.

Oct 9

10

345

— 1 —

O to 4 O to 9

34

10 of 10

1



1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

A horizontal row of ten empty rectangular boxes, each with a vertical line on its left and right sides, intended for handwritten responses.

0 1 2 3 4 5 6 7 8 9

3

4

0 1 2 3 4 5 6 7 8 9

A horizontal line with ten evenly spaced vertical tick marks, extending from approximately x=108 to x=890.

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

a[1][3][2]

```
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