

C Language

Arrays



Saurabh Shukla (MySirG)

Agenda

- ① Introduction to Arrays
- ② Array declaration rules
- ③ Bound Checking
- ④ Sorting
- ⑤ Function call by passing arrays

Introduction to Arrays

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

Array is a group of elements

Write a program to calculate average of
100 numbers.

index (Subscript value)

→ 0 1 2 3 4

99

int a[100];



a[0]



[] subscript operator

a[1] = 25;

```
int a[100], i, s ;  
float avg ;  
printf("Enter 100 numbers");  
for (i=0; i<=99; i++)  
    scanf("%d", &a[i]);  
  
for (i=0, s=0; i<=99; i++)  
    s = s + a[i];  
  
avg = s / 100.0;  
printf("Average is %.f", avg);
```

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.

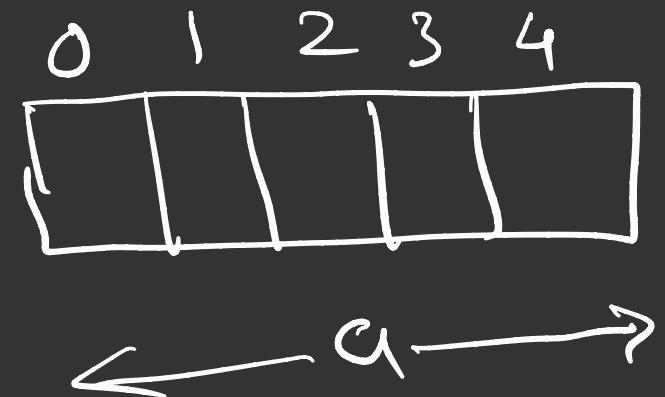
⑦ During declaration you can leave [] empty
only when you initialize array at the same time.

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

Bound Checking

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

```
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

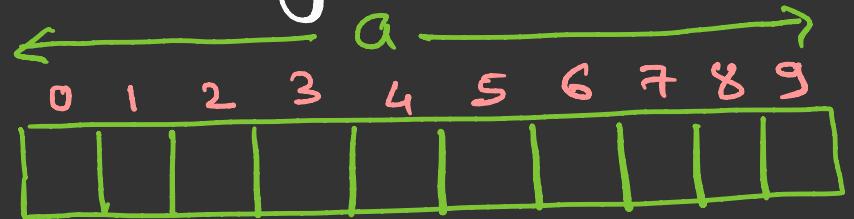
given array →

20 50 90 60 70 80 30 10

Sorted array → 10 20 30 50 60 70 80 90

Function call by passing array

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



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

