

# Arrays

By Mubasshira Ali

# Objective

- ▶ Introduction to Arrays
- ▶ Declaration and Initialization of Arrays
- ▶ Array Operations: Accessing Elements, Array Manipulation
- ▶ Arrays and functions
- ▶ Types of Arrays

# What is Array

An Array is a Linear Data Structure, which is a finite collection of same data items stored in consecutive memory locations.

Syntax:

```
<Datatype><Name>[Size];
```

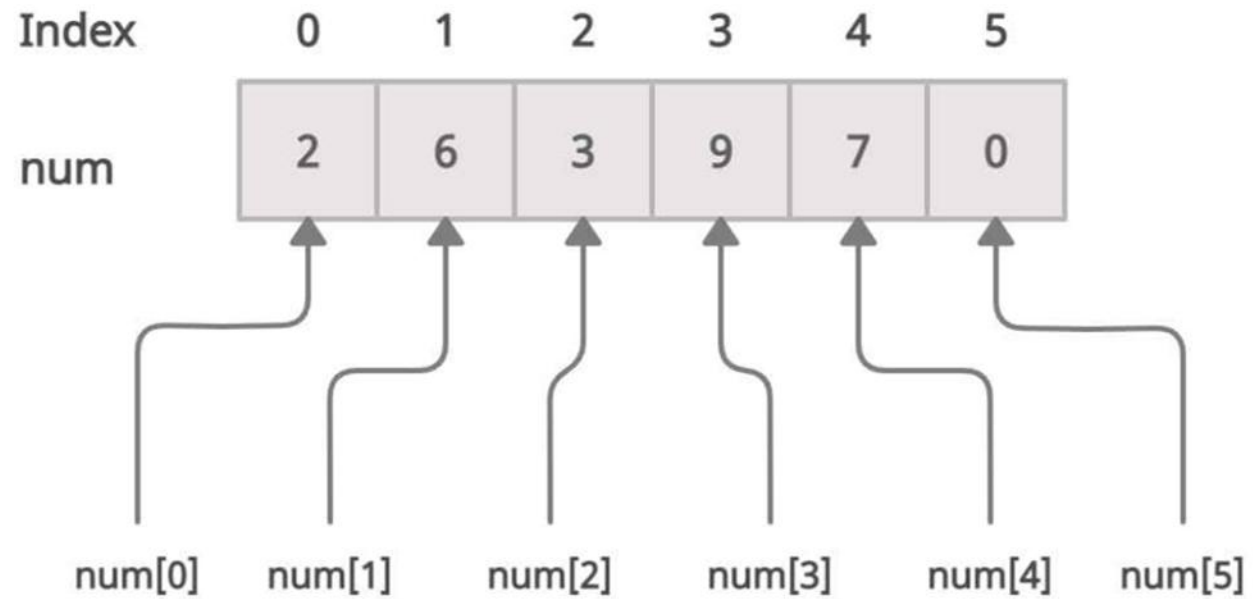
```
<Datatype><Name>[Size]={Elements};
```

Example:

```
int arr[5];
```

```
int arr[5]={1,2,3,4,5};
```

# Memory Representation



# Declaration and Initialization

- ▶ #Define Size 5

```
int num[Size];
```

- ▶ float num[] = {2.5, 7.1, 8.3, 1, 9.1};

```
int a[100] = {0};
```

- ▶ Individual element can be initialized as:

- ▶ num[0] = 2;

- ▶ num[2] = 7;

- ▶ num[5] = 9;

- ▶ int num[2];  
num[2] = {3, 9};



```
int a[] = {1, 2, 3};  
int b[];  
b = a;
```



- ▶ const int Size = 5  
int num[size];



# Example Program:

```
#include<stdio.h>
int main(){
    int num[6]={2,6,3,9,7,0};
    printf("%d\n",num[6]);
    return 0;
}
```

# Operation on Array

Array provide random access to its element using the position of the element called index.

Example:

Array-name[index];

Index always Starts with 0 and ends with Size-1

Processing array can be done using loops (FOR loop)

# Example Program:

```
#include<stdio.h>
```

```
int main() {
```

```
    int age[5]={43,27,35,30,25};
```

```
    for(int i=0; i<5; i++){
```

```
        printf("Employee %d age:%d \n", i+1, age[i]);
```

```
    }
```

```
    return 0;
```

```
}
```



# Example Program:

```
#include<stdio.h>
int main(){
    int sum=0;
    int array[10];
    for(int i=0; i<10; i++){
        scanf("%d",&array[i]);
    }
    for(int i=0; i<10; i++){
        sum+=array[i];
    }
    printf("Sum:%d",sum);
    return 0;
}
```

# Arrays and Function

## ► Passing Individual element to a function

```
#include<stdio.h>
```

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

```
Int main(){  
    int a[5] = {10,20,30,40,50}  
    foo(a[5]);  
    return 0;  
}
```

## ► Passing whole array to a function

```
#include<stdio.h>
#define SIZE 5
void Foo(int a[],int size){
    For(i=0; i<size; i++){
        printf("%d",a[i]);
    }

int main (){
    int a[SIZE]={10,20,14,18,30};
    foo(a,SIZE);
    return 0;
}
```

# Type of Arrays

- ▶ One Dimensional Array
- ▶ Multi Dimensional Array
  - ▶ Two-Dimensional Array
  - ▶ Three-Dimensional array

# Multi Dimensional Array

An Array having two or more then two subscript(index) is called multi dimensional array.

Multi dimensional arrays are just a logical abstraction above a linear storage.

Syntax :

`<Datatype><Name>[s1][s2][s3]..[sn]`

Example:

```
int matrix [2][2]; //2 dimensional
```

```
int table[3][2][2];    //3 dimensional
```

```
char matrix[4][2][3][2];
```

# Two Dimensional Array

A two dimensional stores data as a data as a logical collection of **Rows** and **Columns**

Syntax:

```
<Datatype><Name>[row][column]={elements};
```

Example:

```
char letter[2][3]={{'f','g','h'}, //row 0  
                  {'q','w','e'}} //row 1
```

# Memory Representation of two Dimensional

```
int num[3][4] = {  
    {1, 2, 3, 4},  
    {5, 6, 7, 8},  
    {9, 10, 11, 12}  
};
```

|          |   | col → |    |    |    |
|----------|---|-------|----|----|----|
|          |   | 0     | 1  | 2  | 3  |
| row<br>↓ | 0 | 1     | 2  | 3  | 4  |
|          | 1 | 5     | 6  | 7  | 8  |
|          | 2 | 9     | 10 | 11 | 12 |

|         |           |      |      |      |           |      |      |      |           |      |      |      |
|---------|-----------|------|------|------|-----------|------|------|------|-----------|------|------|------|
|         | ← row 0 → |      |      |      | ← row 1 → |      |      |      | ← row 2 → |      |      |      |
| value   | 1         | 2    | 3    | 4    | 5         | 6    | 7    | 8    | 9         | 10   | 11   | 12   |
| address | 1000      | 1002 | 1004 | 1006 | 1008      | 1010 | 1012 | 1014 | 1016      | 1018 | 1020 | 1022 |

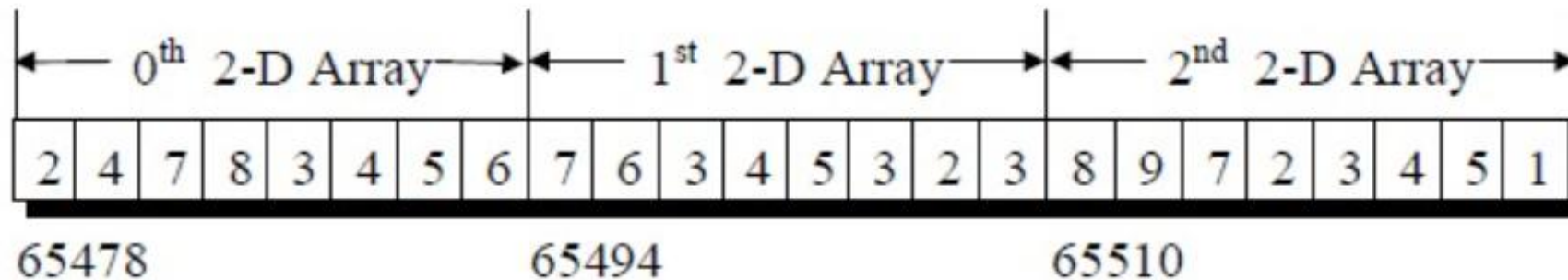
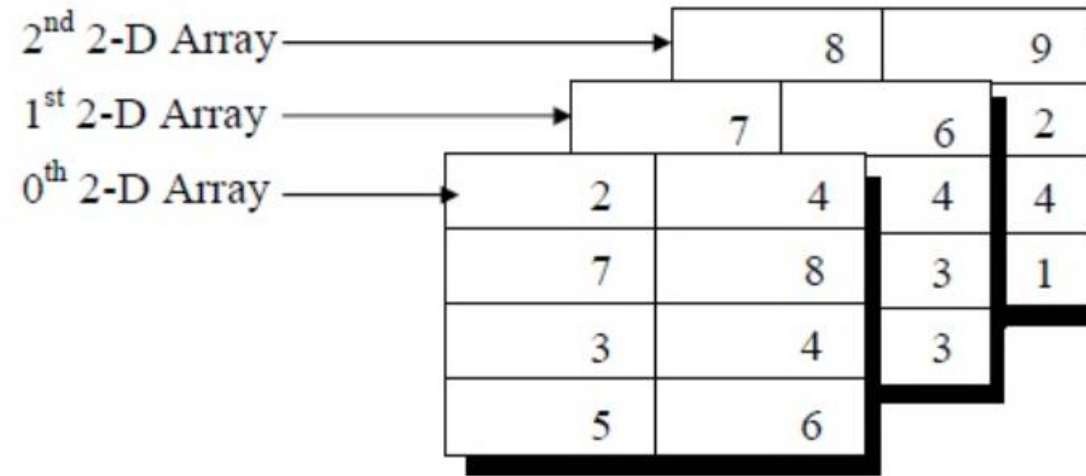
↑

first element of the array num

# Memory Representation of Three dimensional

int num [3][4][2] = {

{ {2,4},{7,8},{3,4},{5,6} },  
{ {7,6},{3,4},{5,3},{2,3} },  
{ {8,9},{7,2},{3,4},{5,1} }  
};





WAP to find the largest number in an array of size 20.

WAP to manage and analyze student marks for a class. the program should have following tasks:

- \* Number of student are 10
- \* Number of subjects are 4
- \* prompt user to enter marks of each student of each subject.
- \* Display the entire mark sheet in tabular format.

Marks Entered

```
-----  
student | Subject 1 | Subject 2 |  
-----|-----|-----|  
1      | 12      | 15      |  
2      | 20      | 17      |  
3      | 10      | 12      |
```

- \* Display subject number and highest marks scored.

Highest Marks in Each subject

subject 1: 20  
subject 2: 16

Thank You