Arrays

Arrays

Definition of Array

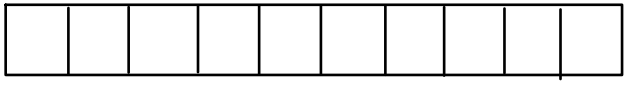
An array is a "Data structure" Containing a "number of data values" (all of which are of "same type")

What is a Data Structure

Data structure is a format for organizing and storing data.

→ Also, each data structure is designed to organize data to suite a specific purpose.

Example: Array is a data structure which you can visualize as follows.



Imagine an array as a large chunk of memory Divided into smaller block of memory and each block is capable of storing a data value of some type.

A	5	6	10	-1	7	8	101	97	88	67

This array Consists of 10 Data Values

A

5	6	10	13	56	76	1	2	4

✓

B

'a'	'b'	'c'	'd'	'e'	'f'

✓

C

a	b	1	1.2	e	34	2	2.5	6

✗

Declaration and Definition of 1D Array

Syntax: data_type name of the Array [no of elements];

Example :- int Arr[5];

Arr

Compiler will allocate a Contiguous block of memory of Size = 5 * sizeof(int)

* The length of an array can be Specified by any Positive integer Constant expression.

```
int A[5];
```
✓

```
int A[5+5];
```
✓

```
int A[5x3];
```
✓

```
int a;
int A[a= 10/3];
```
✓

```
int A[-5];
```
✗

⇒ Specifying the length of an array Using macro is Considered to be an excellent Practice.

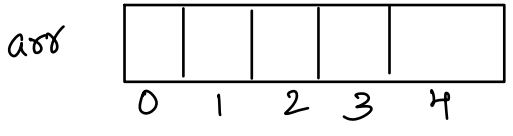
```
#define N 10
int A[N];
```

← This is macro

Accessing Elements from 1D Array

To access an array elements, Just write

```
array_name [index];
```



Accessing the first element of an array : arr[0]

Accessing the Second element of an array : arr[1]

and so on....

⇒ Advantage of Using macro's

How to initialize one dimensional array ?

Method 1:-

```
int A[5] = { 1, 5, -1, 0, 7 };
```

Method 2:-

```
int A[] = { 1, 5, -1, 0, 7 };
```

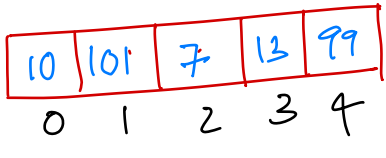
Method 3:-

```
int A[5];
A[0] = 1;
A[1] = 5;
A[2] = -1;
A[3] = 0;
A[4] = 7;
```

Method 4:-

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

10 101 7 13 99



Q what if number of elements are lesser than the length specified?

```
int A[10] = {7, -1, 2, 4, 0, 7}
```

The remaining locations of the array are filled by value 0.

```
int A[10] = {7, -1, 2, 4, 0, 7, 0, 0, 0, 0};
```

A simple tip

```
int A[10];
for (i=0; i<10; i++){
    A[i] = 0;
}
```

→

```
int A[10] = {0};
```

✓

```
int A[10] = { }; ?
```

✗

Because, this is illegal.

=> you must have to specify at least 1 element if can not be completely empty.

=> and it is also illegal to add more elements than the length of an array.

Designated Initialization of Array

Sometimes we want Something like this

```
int A[10] = {1,0,0,0,0,2,3,0,0,0}
```

we want 1 in index 0

2 in index 5

3 in index 6

```
int A[10] = {[0]=1, [5]=2, [6]=3};
```

designated initialization

And each number in Square bracket is said to be a designator.

Introduction to TWO - Dimensional (2D) Array

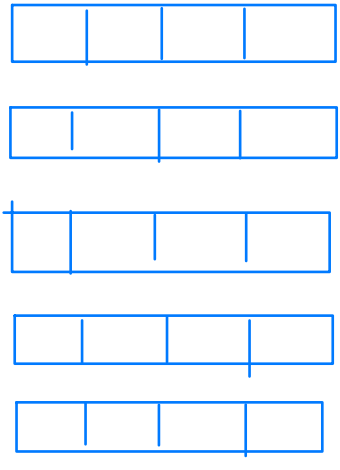
Recall that a multidimensional array is an array of arrays

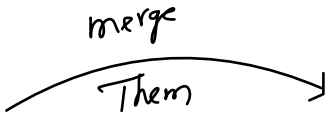
How to declare a 1D array?

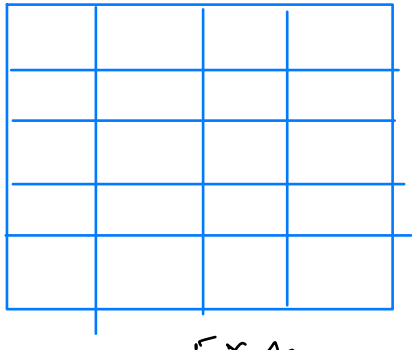
```
int arr[5];
```

Now we add one more dimension here it becomes 2D array

```
int arr[5][4];
```







5 X 4
(rows) (col)

So we can say 2D Dimension array as a matrix

it will have total $5 \times 4 = 20$ elements

The size of this matrix is 20×4

$= 80$ bits

How to Initialize two dimensional array?

Method 1

```
int a[2][3] = {1, 2, 3, 4, 5, 6};
```

	0	1	2
0	1	2	3
1	4	5	6

But there is Some Confusion

Method 2

row 0 row 1

```
int arr[2][3] = {{1,2,3}, {4,5,6}};
```

Col 0 Col 1 Col 2

row 0 →	1	2	3
row 1 →	4	5	6

How to Access 2D Array Elements ?

Using row index and column index

Example

we can access elements stored in 1st row and 2nd column of below array

	0	1	2
0	1	2	3
1	4	5	6

a[0][1]

How to Print 2D array Elements ?

1D array elements can be Printed by Using Single for loop

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

for (i=0; i<5; i++){
        printf ("%d", a[i]);
}
```

2D array elements can be printed using two nested for loops.

```
int a[2][3] = {{1,2,3},{4,5,6}};

for(i=0; i<2; i++){
    for(j=0; j<3; j++){
        printf("%d", a[i][j]);
    }
}
```