

Need of an Array

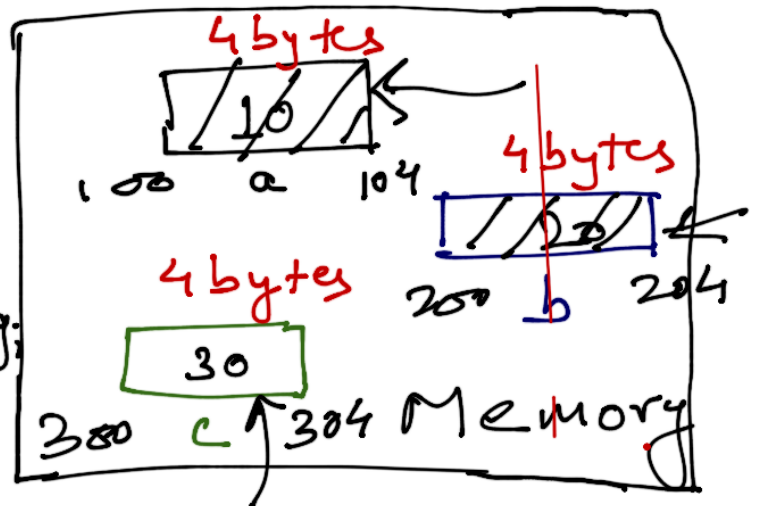
Example

int a;

int b;

int c; float Avg;

$$\text{Avg} = \frac{a+b+c}{3}$$



Instead of engaging so many variables, better to go for one identifier & organize the data.

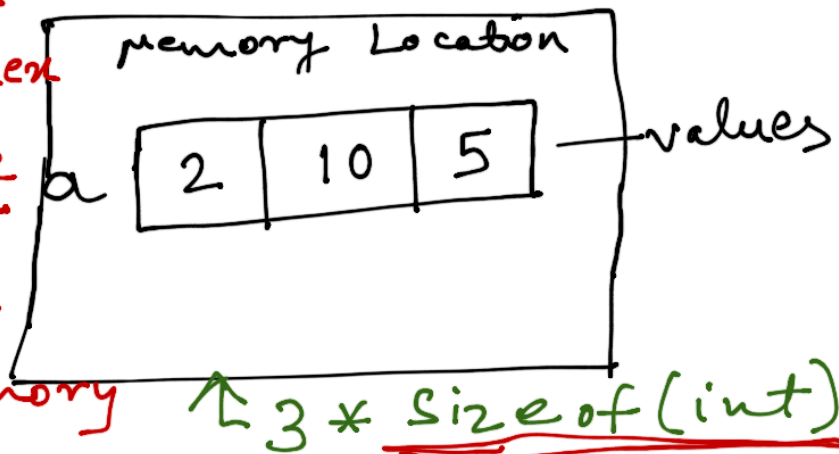
Data type \rightarrow int \downarrow Name of the array a[3] \uparrow No. of elements

- * Series of elements
- * accessible by an index

* Same data type

* unique identifier

* contiguous memory $\uparrow 3 \times \text{Size of (int)}$



Are these Arrays ??

a

1	11	7	56	29	99
---	----	---	----	----	----

b

c	d	e	f	g	h
---	---	---	---	---	---

c

1	d	7	f	29	h
---	---	---	---	----	---

Length of an array

→ It can be specified by any

integer constant expression.

Correct / Incorrect

int a[5];

int a[5+5];

int a[5*3];

int a[-5];

int a, int b[a=22/3]

→ Specify length of array using macro.

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

Initialization of Arrays

Method 1

```
int a[5] = {1, 2, 5, 30, 32};
```

Method 2

```
int a[] = {1, 2, 5, 30, 32};
```

Method 3

```
int a[5];
```

```
a[0] = 1;
```

```
a[1] = 2;
```

```
a[2] = 5;
```

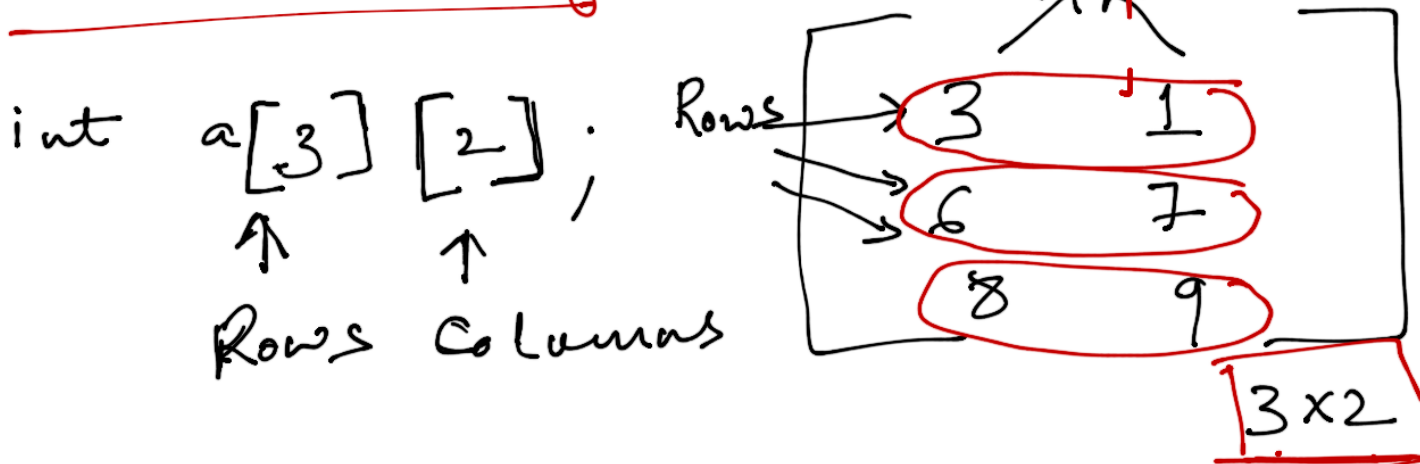
```
a[3] = 30;
```

```
a[4] = 32;
```

Method 4:

```
int a[5];  
for (i = 0; i < 5; i++)  
{  
    cout << "Enter the value";  
    cin >> a[i];  
}
```

2-D Array



Array of Arrays

int a[3][2] = {3, 1, 6, 7, 8, 9}

↓ i j →

	0	1
0	a[0][0]	a[0][1]
1	a[1][0]	a[1][1]
2	a[2][0]	a[2][1]

```
int a[3][2] = { {3,1}, {6,7}, {8,9} };  
int a[][2] = {  
int a[3][] = X..
```

```
int a[3][2], i, j;  
cout << "Enter the elements";  
for (i=0; i < 3; i++)  
{  
    for (j=0; j < 2; j++)  
    {  
        cout << "Enter a element";  
        cin >> a[i][j];  
    }  
}
```

Diagram illustrating the memory layout of the 2D array `a` (3 rows, 2 columns) with indices `i` and `j` shown in red:

	<code>i</code>	<code>j</code>	<code>0</code>	<code>1</code>
<code>a</code>	<code>0</code>		3	1
	<code>1</code>		6	7
	<code>2</code>		8	9

1) Row - Major

2) Column - Major

Row Major

0th Row

1st Row

2nd Row

3 ¹⁰⁰	1 ¹⁰⁴
6 ¹⁰⁸	7 ¹¹²
8 ¹¹⁶	9 ¹²⁰

	3	1	6	7	8	9	
	100	104	108	112	116	120	

Column Major

	3	6	8	1	7	9	
	100	104	108	112	116	120	

	3 ¹⁰⁰	1 ¹⁰⁴
	6 ¹⁰⁸	7 ¹¹²
	8 ¹¹⁶	9 ¹²⁰

Accessing Elements in an Array

for 1-D Array

0	1	2	3
4	10	56	16

$$\text{Base Address}_i = \text{Base Address}_0 + i \times \text{Size of data type}$$

$$= 100 + 2 \times 4$$

for

$$\underline{i=2} = 100 + 8$$

$$= 108$$

for 2-D Array

1) Row-Major

$$\text{Base Address}_{ij} = \text{Base Address}_0 + ((i \times n) + j) \times \text{Size}$$

where, $m \times n$ represents size of the array

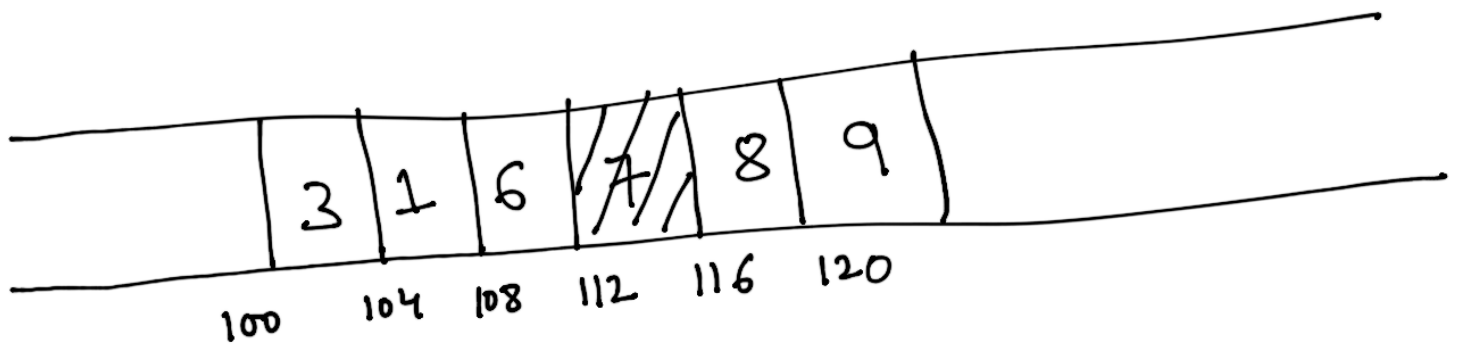
2) Column - Major

$$\text{Base Address}_{ij} = \text{Base Address}_0 + ((j \times n) + i) \times \text{Size}$$

Example:

Row Major

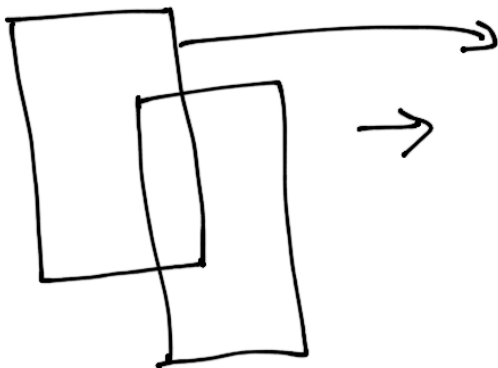
		$a[0][1]$
$a[0][0]$	3	1
$a[1][0]$	6	7
$a[2][0]$	8	9
		$a[2][1]$
		3x2



Address of $a[1][1]$??

$$\begin{aligned}
 \text{Base Address}_{11} &= \text{Base Address}_0 + ((i \times n) + j) \times \text{Size} \\
 &= 100 + [(1 \times 2) + 1] \times 4 \\
 &= 100 + (3) \times 4 \\
 &= 112
 \end{aligned}$$

Example : 3D Array $\rightarrow [0][0][1]$



1	4
10	9
68	99

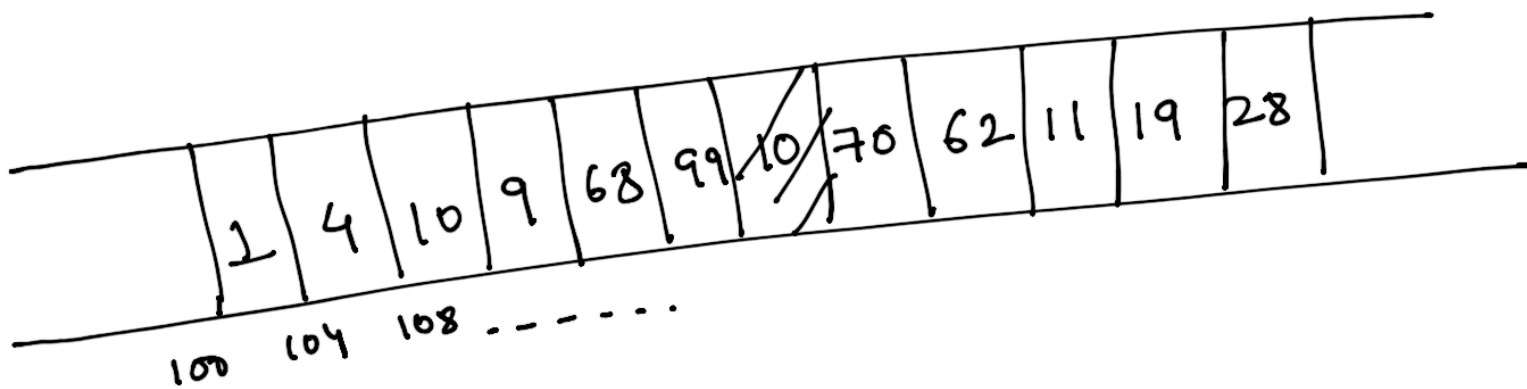
1st plane

10	70
62	11
19	28

2nd plane

A [3][2][2]

Row Major



$A[i][j][k]$

$$\text{Base Address}_{[0][0][1]} = \text{Base Address}_{[0][0][0]} + [(n \times i) + j + (m \times n \times k)] \times \text{size}$$

$$= 100 + [(2 \times 0) + 0 + (3 \times 2 \times 1)] \times 4$$

$$= 100 + [6] \times 4 = 124$$