1. *Memory Representation of Arrays*

Array allocate a memory location and each location's size is the size of the data type.

Suppose it is integer which is of 4 bytes and arrays byte is 5 then it holds 5 adjacent contiguous memory location with indexes.

In Single Dimensional Array:

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

4 bytes	4bytes	4bytes	4bytes	4bytes
$elem = 1$	$elem = 2$	$elem = 3$	$elem = 4$	$elem = 5$
0	1	2	3	4

$$arr[0] = 1$$

$$arr[1] = 2$$

$$arr[2] = 3$$

$$arr[3] = 4$$

$$arr[4] = 5$$

Here elements $= 1, 2, 3, 4, 5$ ***and index*** $= 0, 1, 2, 3, 4.$

And each index is sized 4 bytes each $= 20$ ***bytes.***

Two-Dimensional Array:

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

where $, row = 3$ ***and column*** $= 4.$

$elem = 1$ [0][0]{4 bytes}	$elem = 2$ [0][1]{4 bytes}	$elem = 3$ [0][2]{4bytes}	$elem = 4$ [0][3]{4bytes}
$elem = 5$ [1][0]{4bytes}	$elem = 6$ [1][1]{4 bytes}	$elem = 7$ [1][2]{4 bytes}	$elem = 8$ [1][3]{4 bytes}
$elem = 9$ [2][0]{4 bytes}	$elem = 10$ [2][1]{4 bytes}	$elem = 11$ [2][2]{4 bytes}	$elem = 12$ [2][3]{4 bytes}

$$Hence\ Total\ Memory\ taken\ by\ the\ Array$$

$$= 4 \times 3 \times 4bytes = 48\ bytes.$$

Three-Dimensional Array:

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

$$where\ , page\ size = 2, row\ = 3\ and\ column\ = 4.$$

Page 1

$elem = 1$ [0][0][0]{4 bytes}	$elem = 2$ [0][0][1]{4 bytes}	$elem = 3$ [0][0][2]{4bytes}	$elem = 4$ [0][0][3]{4bytes}
$elem = 5$ [0][1][0]{4bytes}	$elem = 6$ [0][1][1]{4 bytes}	$elem = 7$ [0][1][2]{4 bytes}	$elem = 8$ [0][1][3]{4 bytes}
$elem = 9$ [0][2][0]{4 bytes}	$elem = 10$ [0][2][1]{4 bytes}	$elem = 11$ [0][2][2]{4 bytes}	$elem = 12$ [0][2][3]{4 bytes}

Page 2

$elem = 1$ [1][0][0]{4 bytes}	$elem = 2$ [1][0][1]{4 bytes}	$elem = 3$ [1][0][2]{4bytes}	$elem = 4$ [1][0][3]{4bytes}
$elem = 5$ [1][1][0]{4bytes}	$elem = 6$ [1][1][1]{4 bytes}	$elem = 7$ [1][1][2]{4 bytes}	$elem = 8$ [1][1][3]{4 bytes}
$elem = 9$ [1][2][0]{4 bytes}	$elem = 10$ [1][2][1]{4 bytes}	$elem = 11$ [1][2][2]{4 bytes}	$elem = 12$ [1][2][3]{4 bytes}

Total memory it takes: $2 \times 3 \times 4 \times 4bytes = 96\ bytes.$

Note: As it takes adjacent contiguous memory location it looks like this: