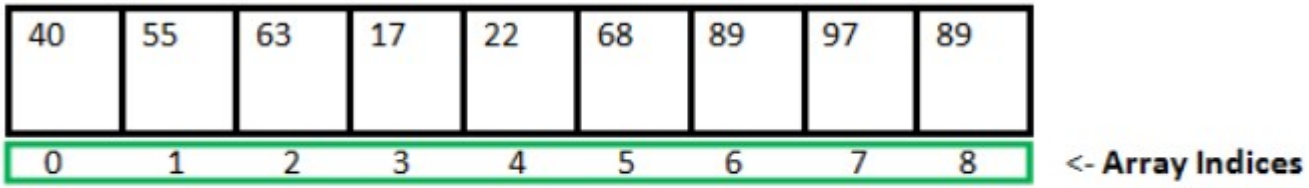
C++

پریسا حامدروح بخش

موسسه ی پارس پژوهان

Arrays



Array Length = 9
First Index = 0
Last Index = 8

Array

1. It is a group of variables of similar data types referred to by a single element.
2. Its elements are stored in a contiguous memory location.
3. The size of the array should be mentioned while declaring it.
4. Array elements are always counted from zero (0) onward.
5. Array elements can be accessed using the position of the element in the array.
6. The array can have one or more dimensions.

Arrays

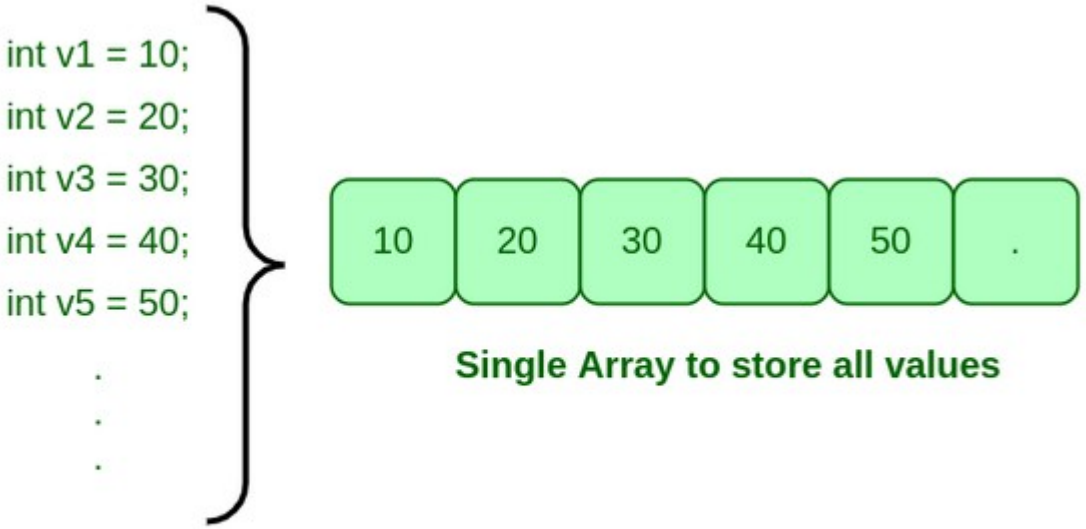
```
int b[5] = {11, 45, 62, 70, 88};
```

- **The values are provided in a comma separated list, enclosed in {curly braces}.**

> The number of values between braces { } must not exceed the number of the elements declared within the square brackets [ ].

Why do we need arrays?

We can use normal variables (v1, v2, v3, ..) when we have a small number of objects, but if we want to store a large number of instances, it becomes difficult to manage them with normal variables. The idea of an array is to represent many instances in one variable.



Advantages/Disadvantages

Advantages:-

- Code Optimization: we can retrieve or sort the data efficiently.
- Random access: We can get any data located at an index position.

Disadvantages:-

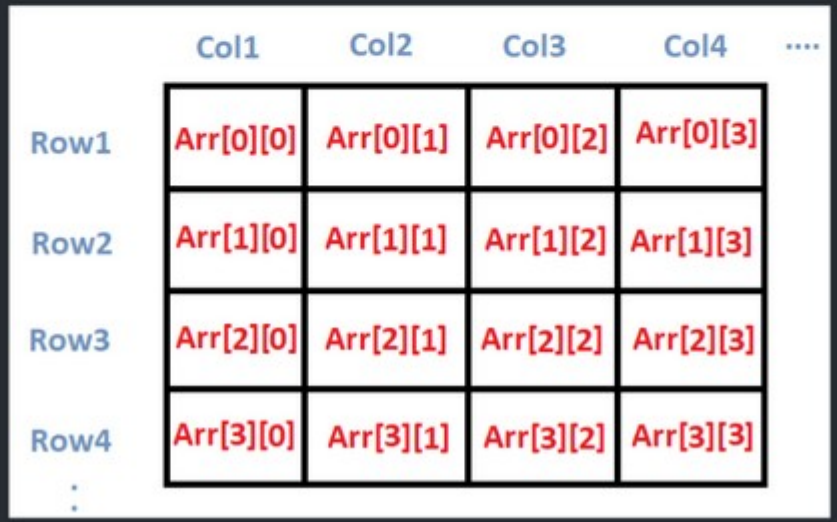
- Size Limit: We can store only the fixed size of elements in the array. It doesn't grow its size at runtime.

2D Array

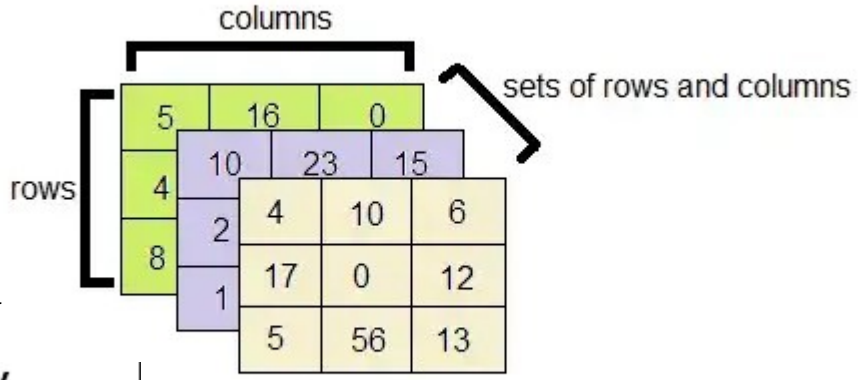
C + +

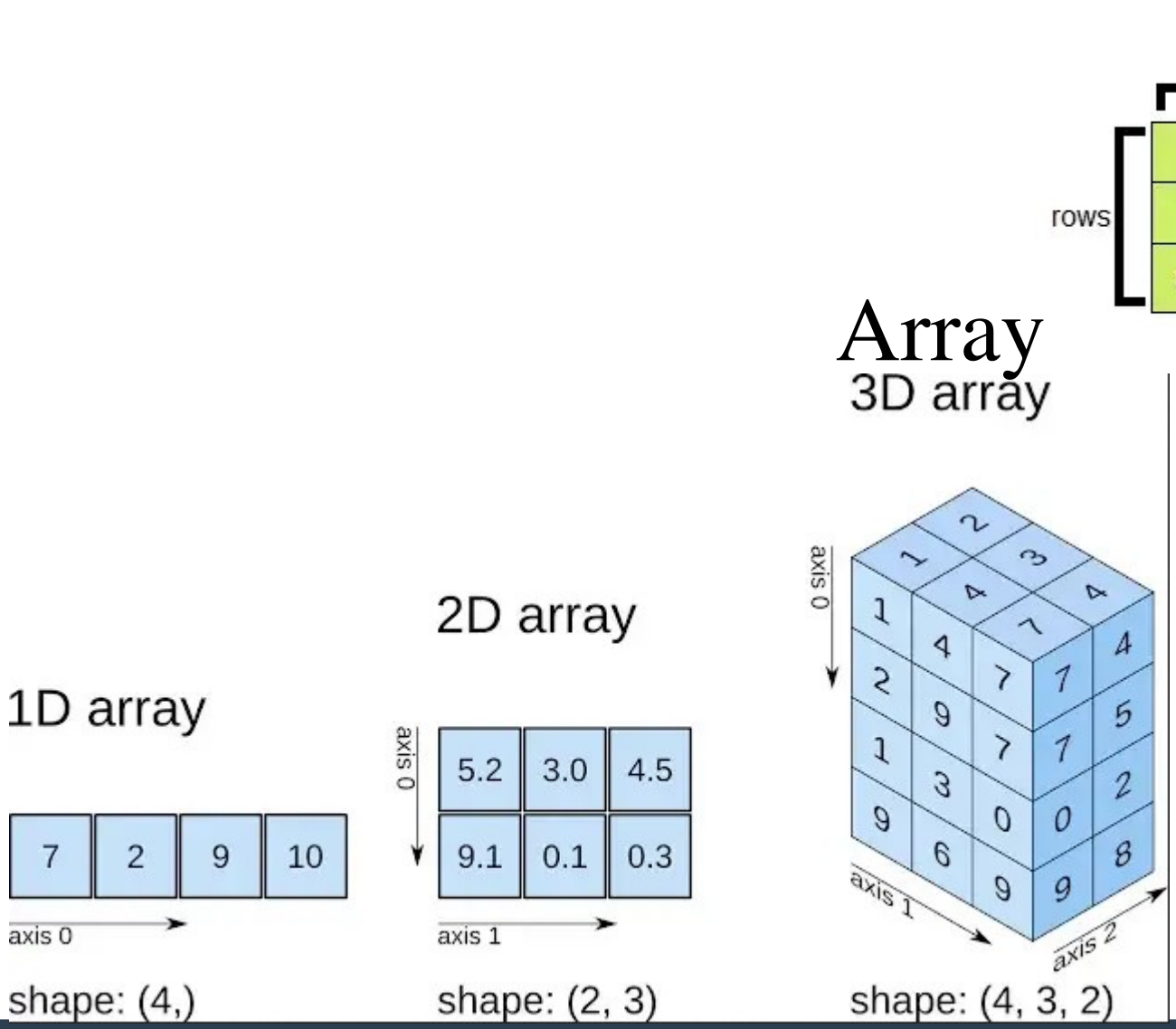
2-Dimensional Arrays

- 2-D Arrays can be defined as an array of arrays,
- It can also represent a Matrix,
- Each element is represented as Arr[row][column], where Arr[][] is the 2D array.



3D Array


columns
sets of rows and columns
rows
5
16
0
4
10
23
15
8
2
4
10
6
1
17
0
12
5
56
13


Array
3D array
1D array
7
2
9
10
axis 0
shape: (4,)
2D array
axis 0
5.2
3.0
4.5
9.1
0.1
0.3
axis 1
shape: (2, 3)
axis 0
axis 1
axis 2
shape: (4, 3, 2)

Question ?