LINKED LIST : INTRO

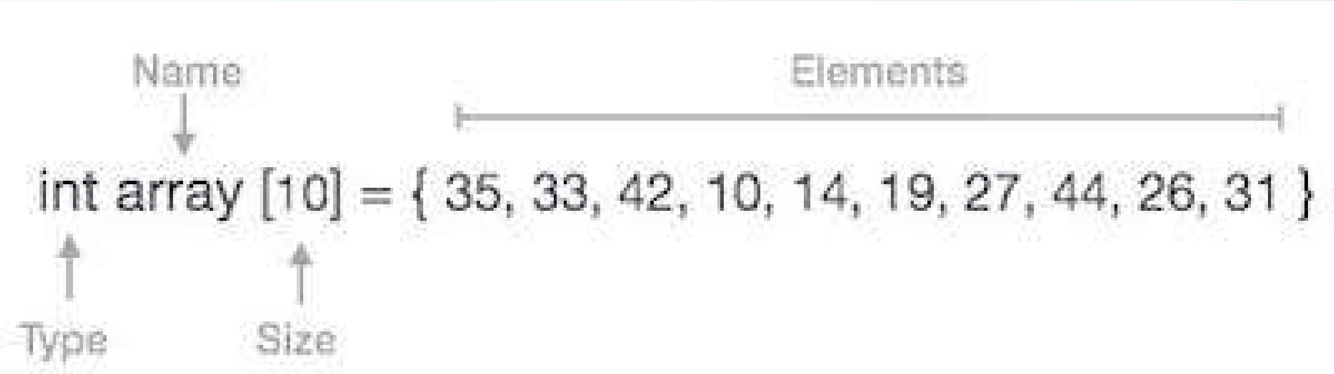
ROMI FADILLAH RAHMAT

ARRAY

- Array is a container which can hold a fix number of items and these items should be of the same type. Most of the data structures make use of arrays to implement their algorithms.
- Following are the important terms to understand the concept of Array.
  - **Element** − Each item stored in an array is called an element.
  - **Index** − Each location of an element in an array has a numerical index, which is used to identify the element.

ARRAY REPRESENTATION

- Arrays can be declared in various ways in different languages. For illustration, let's take C array declaration.
- As the illustration :
  - Index starts with 0
  - Array length / size is 10
  - Each element can be accessed via index.



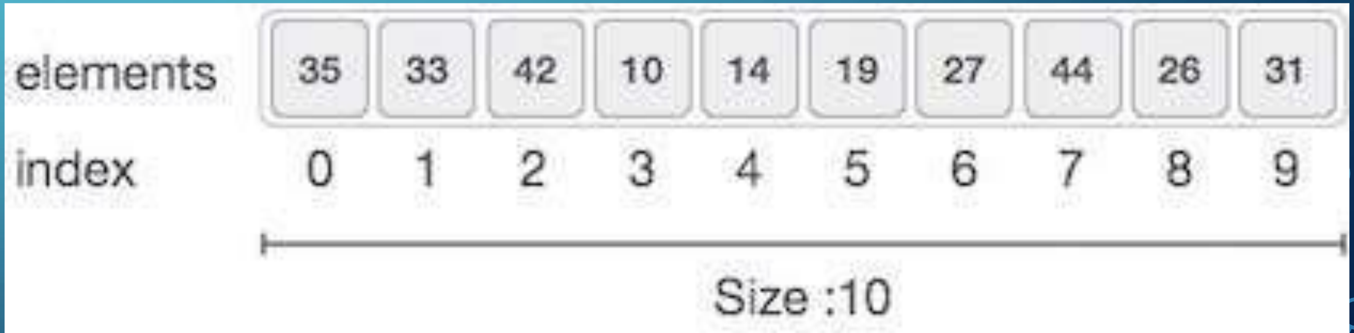
Name

Elements

```
int array [10] = { 35, 33, 42, 10, 14, 19, 27, 44, 26, 31 }
```

Type

Size



elements	35	33	42	10	14	19	27	44	26	31
index	0	1	2	3	4	5	6	7	8	9

Size :10

ADVANTAGES OF ARRAYS

- Simple and easy to use
- Faster access to the elements (constant access)

DISADVANTAGES OF ARRAY

- Fixed Size : The size of the array is static (specify the array size before using it).
- One block allocation : To allocate the array at the beginning itself, sometimes it may not be possible to get the memory for the complete array (if the array size is big).
- Complex position-based insertion : To insert an element at a given position then we may need to shift the existing elements. This will create a position for us to insert the new element at the desired position. If the position at which we want to add an element is at the beginning then the shifting operation is more expensive.

DYNAMIC ARRAYS

- Dynamic array (also called as growable array, resizeable array, dynamic table or array list) is a random access, variable-size list data structure that allows elements to be added or removed.
- Way to implement, initially start with some fixed size array. As soon as that array becomes full, create the new array of size double than the original array. Similarly, reduce the array size to half if the elements in the array are less than half.

WHAT IS LINKED LIST?

- Linked list is a data structure used for storing collections of data. Linked list has the following properties :
  - Successive elements are connected by pointers
  - Last elements points to NULL
  - Can grow and shrink in size during execution of a program
  - Can be made just as long as required (until systems memory exhausts)
  - It does not waste memory space (but takes some extra memory for pointers).

WHY LINKED LIST

- First we need to identify the difference between linked lists and arrays.
- Both Linked Lists and Arrays are used to store collection of Data.
- However, there are a certain condition when arrays are suitable and there are a certain condition when linked lists are suitable.

ADVANTAGE OF LINKED LISTS

- They can be expanded in constant time.
- In array to create array, we must allocate memory for a certain number of elements.
- To add more elements to the array then we must create a new array and copy the old array into the new array. This can take lot of time
- With linked list we can start with space for just one element allocated and add on new elements easily without the need to do any copying and reallocating.

DISADVANTAGE OF LINKED LISTS

- The main disadvantage is access time to individual elements. Array is random-access, which means it takes O*(1)* to access any element in the array. Linked list takes O*(n)* for access to an element in the list in the worst case.