# Module-3: Data Structures

Dr. P. Gayathri
Associate Professor
SCOPE, VIT University

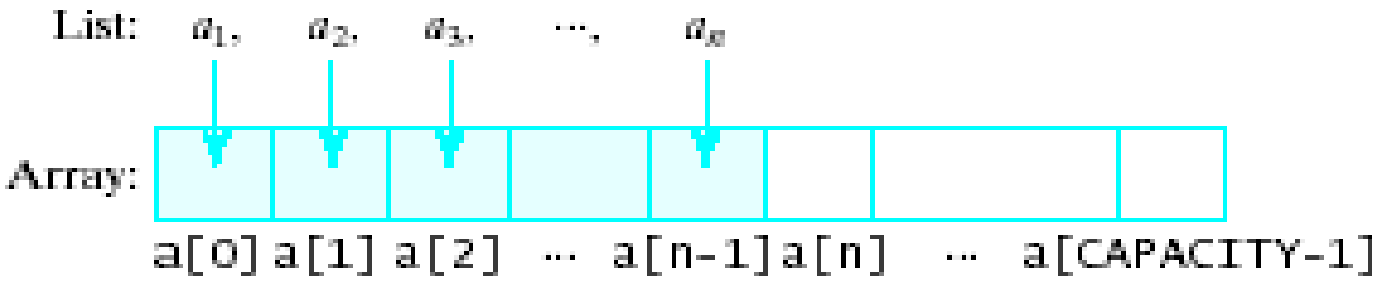
# Arrays

Array is a data structure, which provides the facility to store a collection of data of same type under single variable name.

- Continuous storage structure
- Elements are sequential

# Array-Based Implementation of Lists

- An array is a viable choice for storing list elements
  - Element are sequential
  - Algorithm development is easy
- Normally sequential orderings of list elements.



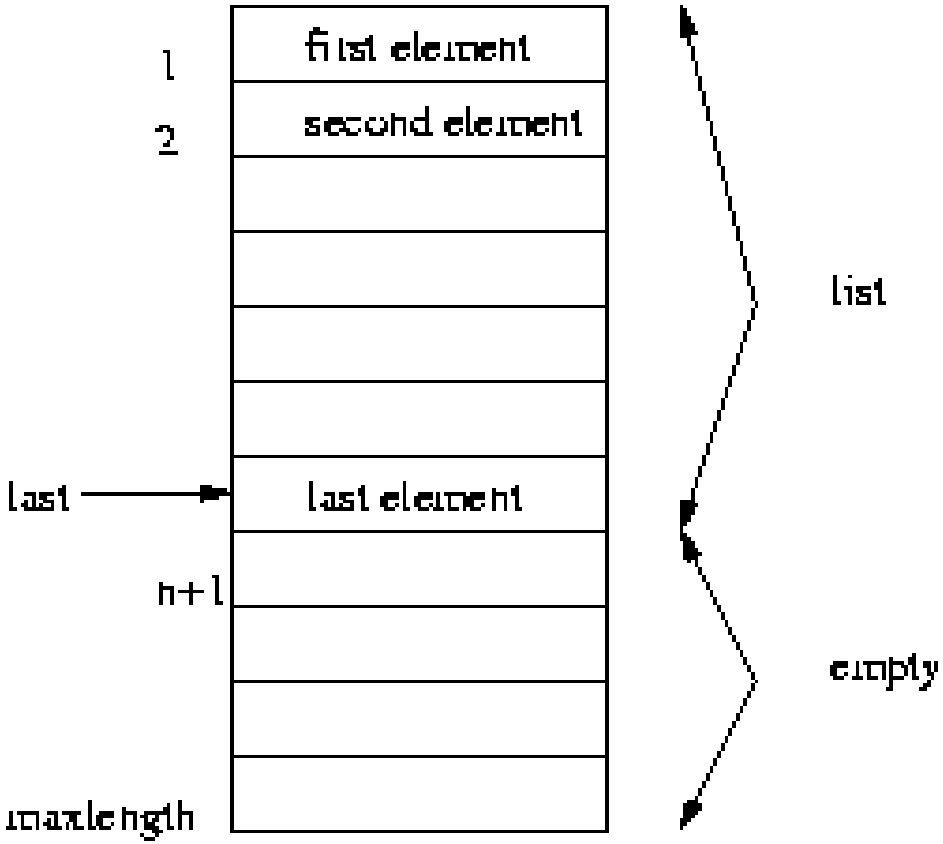
- sequence of zero or more elements

$$A_1, A_2, A_3, \ldots A_N$$

- N: length of the list
- $A_1$: first element
- $A_N$: last element
- $A_i$: position i
- If N=0, then empty list
- Linearly ordered
  - $A_i$ precedes $A_{i+1}$
  - $A_i$ follows $A_{i-1}$

# Array Implementation

- Elements are stored in contiguous array positions

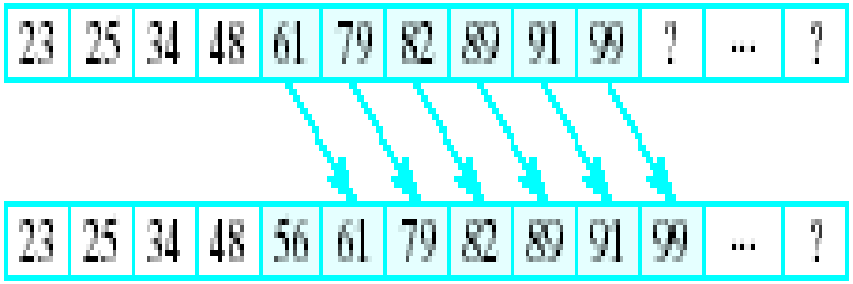


# Operations

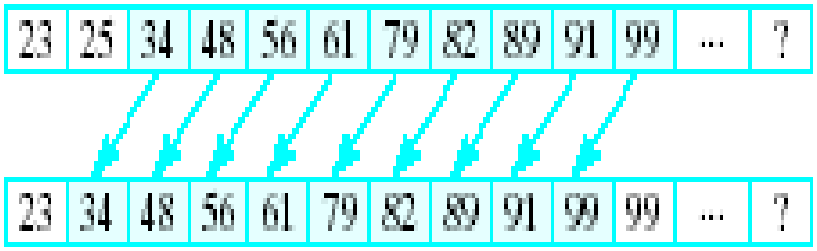
- printList: print the list
- find: locate the position of an elements in a list
  - list: 34,12, 52, 16, 12
  - find(52) $\rightarrow$ 3
- insert: insert an object to a list
  - insert(x,4) $\rightarrow$ 34, 12, 52, x, 16, 12
- remove: delete an element from the list
  - remove(52) $\rightarrow$ 34, 12, x, 16, 12
- findKth: retrieve the element at a certain position

# Operations

- Insert
  - Shift elements to right of insertion point
- Delete
  - Shift elements back




Also adjust **`size`** up or down



# Types

- Ordered list
- Unordered list

# An Array-Based Implementation - Summary

- **Good** things:
  - Fast, random access of elements
  - Very memory efficient
- **Bad** things:
  - Slow deletion and insertion of elements
  - Size must be known when the array is created and is fixed (static)