SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

COURSE NAME : 19CS307 - DATA STRUCTURES

II YEAR / III SEMESTER

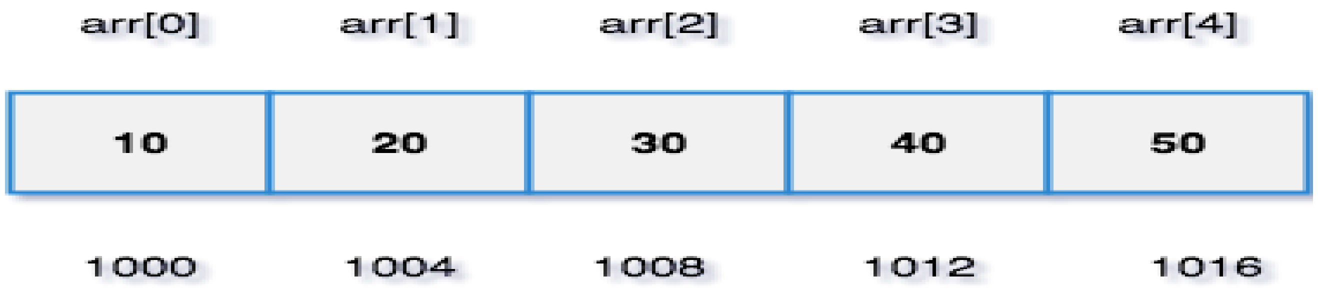
Unit 1- LINEAR DATA STRUCTURES -LIST

Array Based Implementation

ARRAY BASED IMPLEMENTATION OF LIST

- An array is a collection of variables in the same datatype.
- We can’t group different data types in the array. Like, a combination of integer and char, char and float etc.
- Hence array is called as the homogeneous data type.

Ex: **int** arr[**5**]={**10,20,30,40,50**};



- Using index value, we can directly access the desired element in the array.
- Array index starts from 0, not 1.
- To access the 1st element, we can directly use index 0. i.e a[0]
- To access the 5th element, we can directly use index 4. i.e a[4]
- We can manipulate the Nth element by using the index N - 1. {Where N > 0}
- In general, an array of size N will have elements from index 0 to N-1.

Insertion operation

Insert a given element at a specific position in an array.

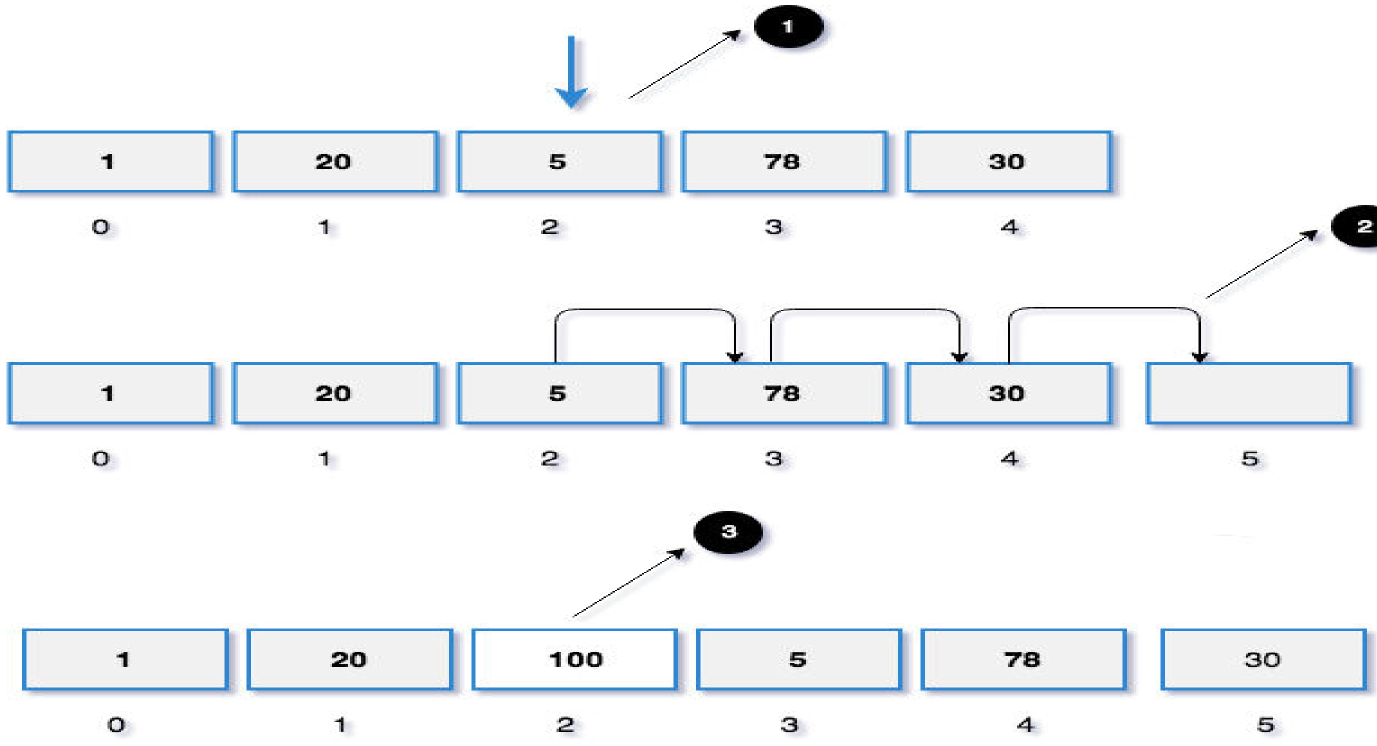
Algorithm

1. Get the **element** value which needs to be inserted.
2. Get the **position** value.
3. Check whether the position value is valid or not.
4. If it is valid,
   - Shift all the elements from the last index to position index by 1 position to the right.
   - insert the new element in arr[position]
5. Otherwise,
   - Invalid Position

Insertion operation

Input

- int arr[5] = {10, 20, 30, 40, 50}
- Element = 100 position = 2.



Delete operation

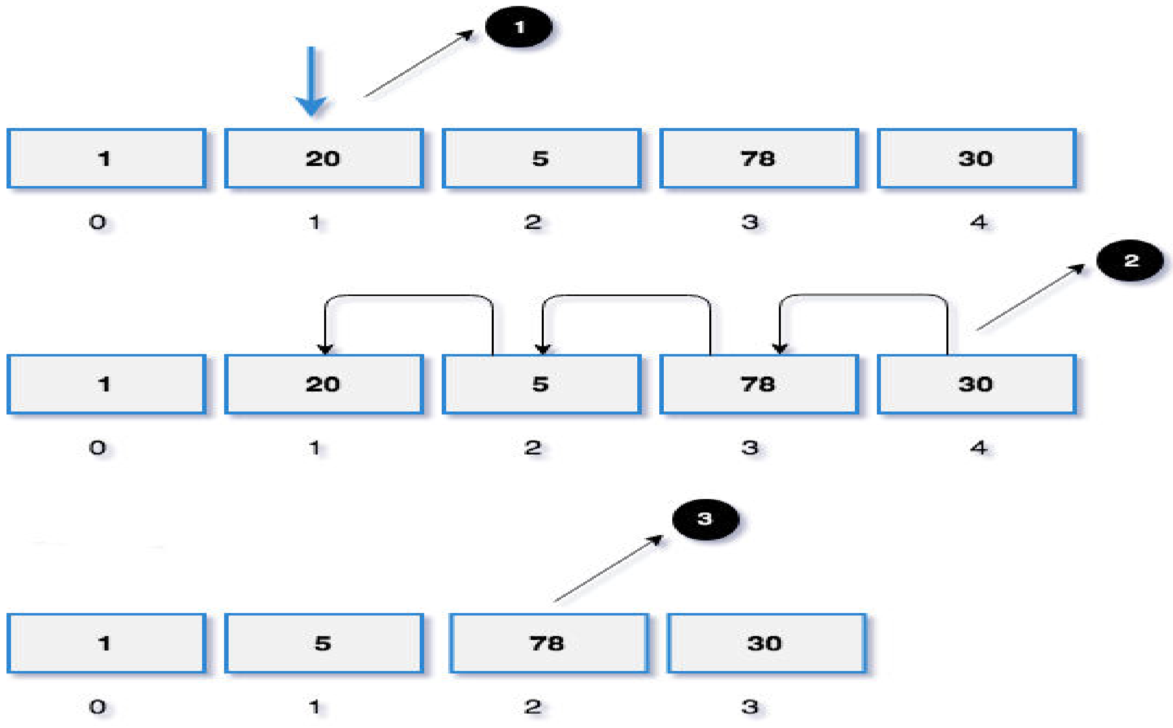
Delete a given element from an array.

Algorithm

1. Find the given element in the given array and note the index.
2. If the element found,
   - Shift all the elements from index + 1 by 1 position to the left.
   - Reduce the array size by 1.
3. Otherwise, print "Element Not Found"

Delete operation

- **Input**
- Array : {1, 20, 5, 78, 30}
- Element : 78



Search operation

Search whether the given key is present or not in the array.

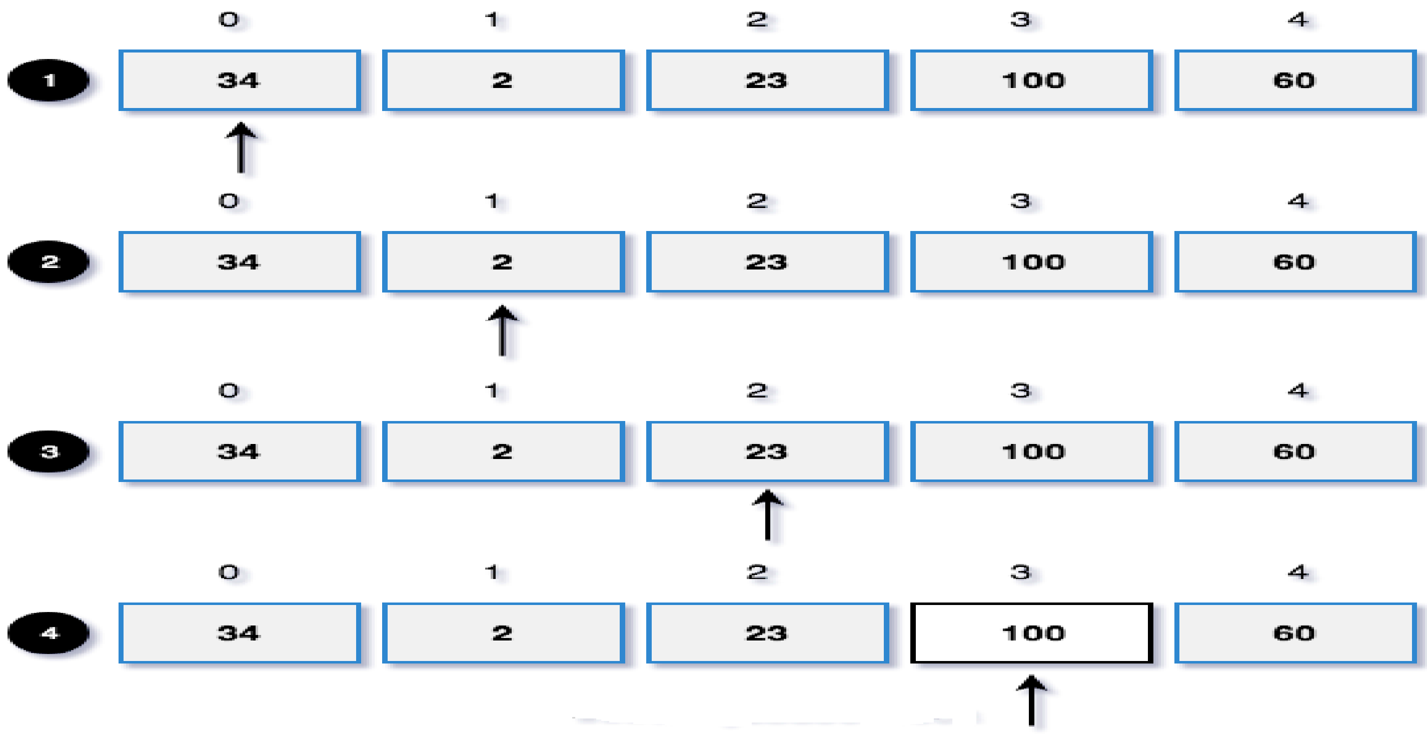
Algorithm

1. Iterate the array using the loop.
2. Check whether the given key present in the array i.e. arr[i] == key.
3. If yes,
   - print "Search Found".
4. Else
   - print "Search Not Found".

Search operation

Input

- arr[5] = {10, 30, 5, 100, 4};
- key = 30



- **Advantages**
  - There is no wasted space for an individual element (do not need space for pointers)
- **Disadvantages**
  - Lacking efficiency for insertion/deletion operations and memory allocation.

Application
  - Arrays are used to implement data structures like a stack, queue, etc.
  - Arrays are used for matrices and other mathematical implementations.
  - Arrays are used in lookup tables in computers.
  - Arrays can be used for CPU scheduling

REFERENCES

1. M. A. Weiss, “Data Structures and Algorithm Analysis in C”, Pearson Education, 8th Edition, 2007. [Unit I, II, III, IV,V]
2. ReemaThareja, “Data Structures Using C”, Second Edition , Oxford University Press, 2011
3. A. V. Aho, J. E. Hopcroft and J. D. Ullman, “Data Structures and Algorithms”, Pearson Education, 2nd Edition, 2007
4. Stephen G. Kochan, “Programming in C”, 3rd edition, Pearson Education
5. A.M.Tenenbaum, Y. Langsam and M. J. Augenstein, “Data Structures using C”,PearsonEducation, 1st Edition, 2003.

THANK YOU