

**NAME : SYED MUHAMMAD SALMAN
KHAN
ID : 13662
SEMESTER: 6TH
SUBJECT : DATA STRUCTURE AND
ALGORITHM
SUBMITTED TO : MUHAMMAD ADIL
DATE 20/8/20**

Q NO 1:#

Algorithm for inserting an element in a 1-d array:

- There must be one location empty available in the array.
- If the new element is to be inserted at the end of the array then there is no problem.
- If the new element is to be inserted at the start or middle of the array ,then all the elements from that positions will have to move one location forward.

Part b:

Find the address of 52nd element of a One Dimensional Array of A[], first element is stored on 54360 and each element occupies 2 memory locations.

Answer:

first element is stored on 54360

each element occupies 2 memory locations

address of 52nd element of a One Dimensional Array of

$A[] = 54360 + (51 \times 2)$

$= 54462$

Address of 52nd element is $= 54462$

Q#2: Design an algorithm to delete an element from a One Dimensional Array.

Dry steps:

1. Item \leftarrow A[K]
2. For (J \leftarrow K to N-1)
 //Start of for loop
(a) . A[J] \leftarrow A[J+1]
 //End of for loop
3. N \leftarrow N-1
4. Exit

Where

- Item is element to be deleted and it is on kth position of A[.]
- J is the centre variable of for loop
- N is the total number of elements presents in A[.]

Q #3 :Create an Algorithm for Binary Search.

Binary search algorithm:

1. Lb <-1
2. Ub <- max
3. Found <- false
4. While (lb <= ub) and (found = false)
//start of while loop
 - (a) mid <- (lb + ub) int-div 2.
 - (b) If (item =A[mid])
Then
Found = true
 - Else
If(item < A[mid])
Then
ub <- mid-1
 - Else
Lb <- mid+1//end of while loop
5. If (found = true)
Then
Display” search successful”
- Else
Display” search unsuccessful”
6. Exit

Where

- Ub = upper bound
- Lb = lower bound
- Found is boolean variable
- Mid is middle of current selected part of A[.]
- Item is element to be searching in A[.]