



SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
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 MCA

19CAT602 – DATA STRUCTURES & ALGORITHMS **I YEAR I SEM**

UNIT I - ELEMENTARY DATA STRUCTURES

TOPIC 2 - Arrays



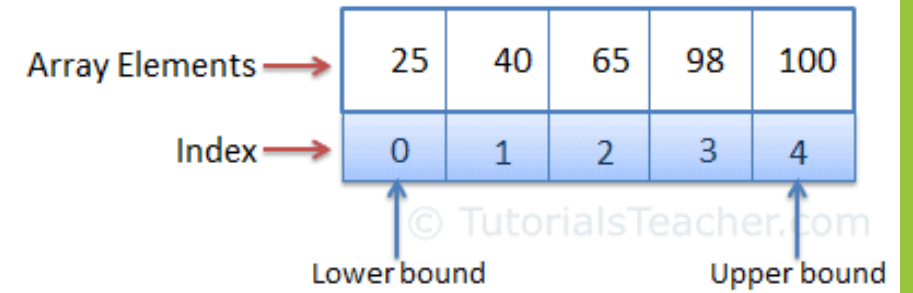
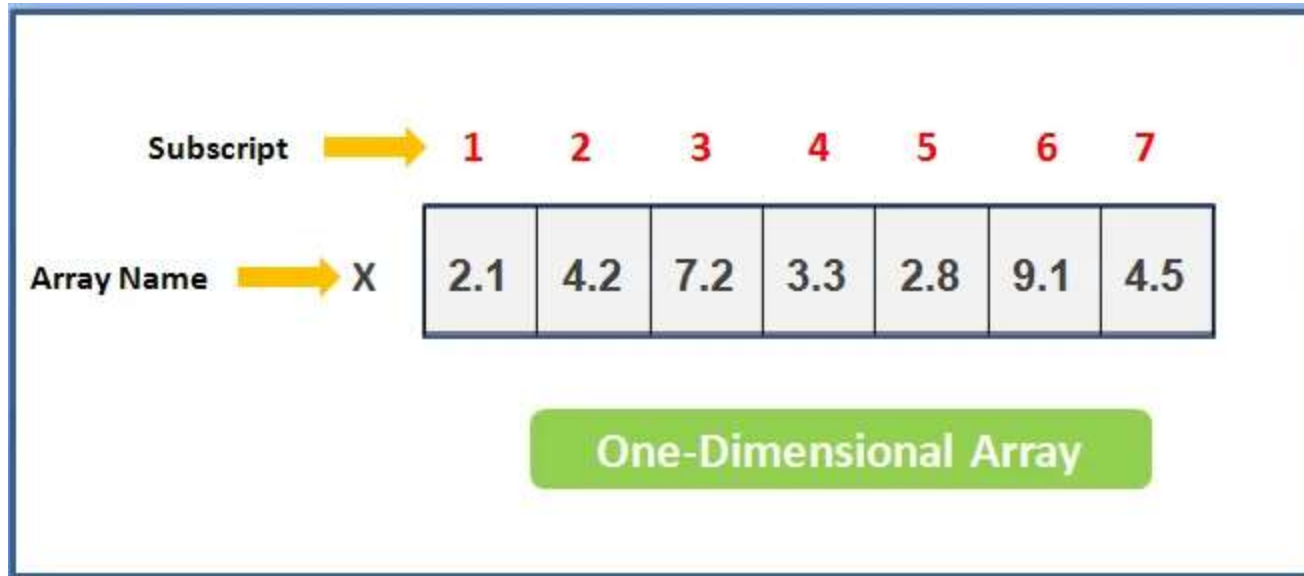
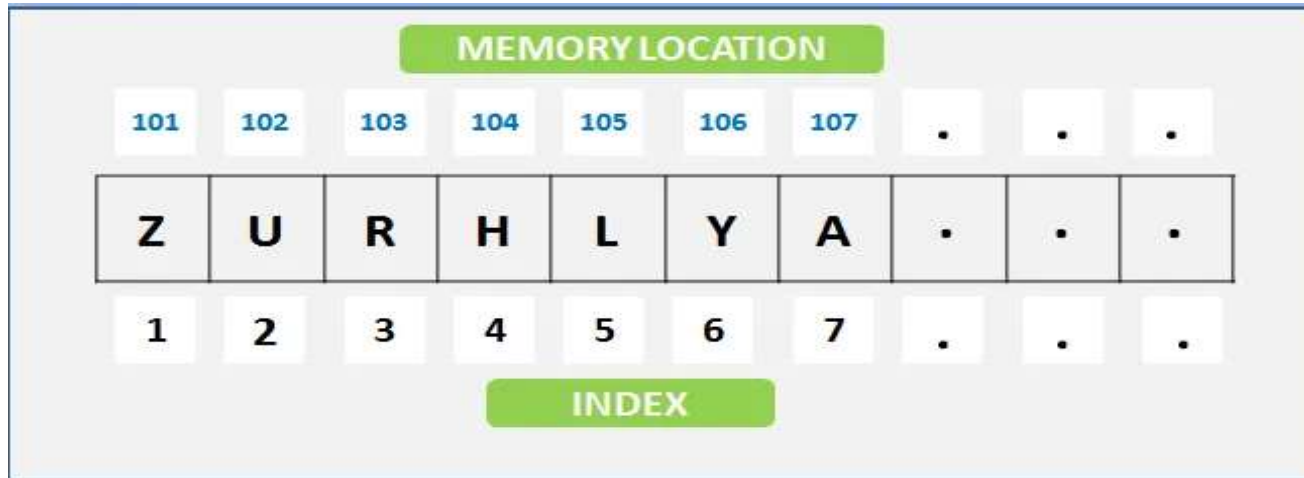
Arrays

An *array* is a data structure that contains a group of elements. Typically these elements are all of the same data type, such as an integer or string.





Arrays



1. **Element** – Each item stored in an array is called an element.
2. **Index** – Each location of an element in an array has a numerical index, which is used to identify the element.



Arrays Types

Index	1	2	3	4	5	6
Value	15	17	25	90	110	221

One - Dimensional Array

Index	1	2	3
1	10	15	7
2	9	25	30
3	39	2	84

Two - Dimensional Array

Index	1	2	3		
1	10	15	7	31	44
2	9	25	30	33	55
3	39	2	84	90	63

Multi - Dimensional Array



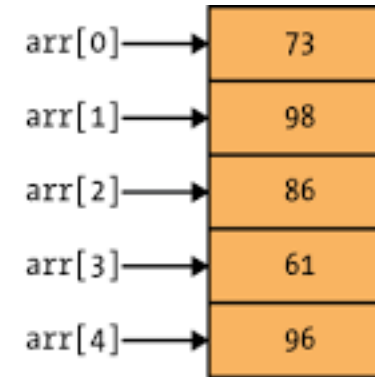
Basic Operations

1. **Traverse** – Print all the array elements one by one.
2. **Insertion** – Adds an element at the given index.
3. **Deletion** – Deletes an element at the given index.
4. **Search** – Searches an element using the given index or by the value.
5. **Update** – Updates an element at the given index.



Traverse Operation

[0]	[1]	[2]	[3]	[4]
73	98	86	61	96



```
#include <stdio.h>
main()
{
    int LA[] = {1,3,5,7,8};
    int n = 5;
    printf("The original array elements are :\n");
    for(i = 0; i<n; i++)
    {
        printf("LA[%d] = %d \n", i, LA[i]);
    }
}
```

The original array elements are :

- LA[0] = 1
- LA[1] = 3
- LA[2] = 5
- LA[3] = 7
- LA[4] = 8



Insertion Operation

```
int LA[] = {2,4,1,7}, item = 10, k = 3, n = 4;
int i = 0, j = n;
n = n + 1;
while( j >= k)
{
    LA[j+1] = LA[j];
    j = j - 1;
}
LA[k] = item;
printf("The array elements after insertion :\n");

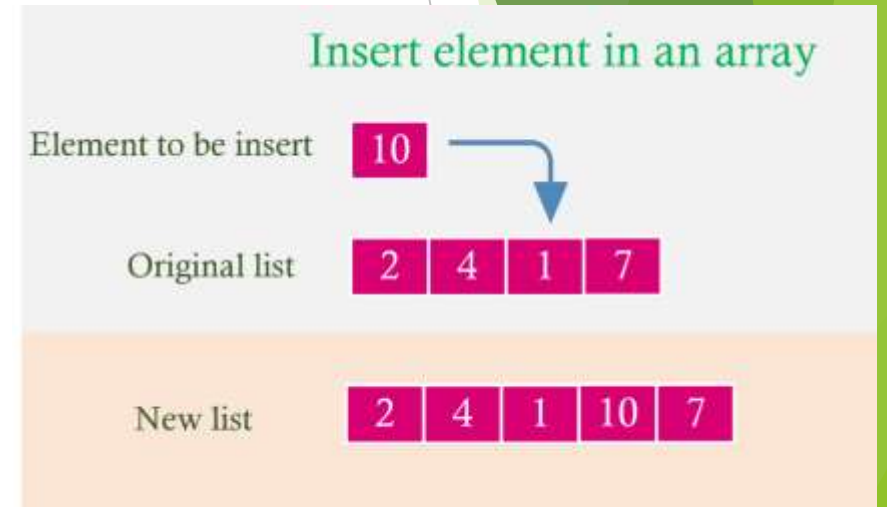
for(i = 0; i < n; i++) {
    printf("LA[%d] = %d \n", i, LA[i]);
}
```

The original array elements are :

- LA[0] = 2
- LA[1] = 4
- LA[2] = 1
- LA[3] = 7

The array elements after insertion :

- LA[0] = 2
- LA[1] = 4
- LA[2] = 1
- LA[3] = 10
- LA[4] = 7





Deletion Operation

1. Start
2. Set $J = K$
3. Repeat steps 4 and 5 while $J < N$
4. Set $LA[J] = LA[J + 1]$
5. Set $J = J + 1$
6. Set $N = N - 1$
7. Stop

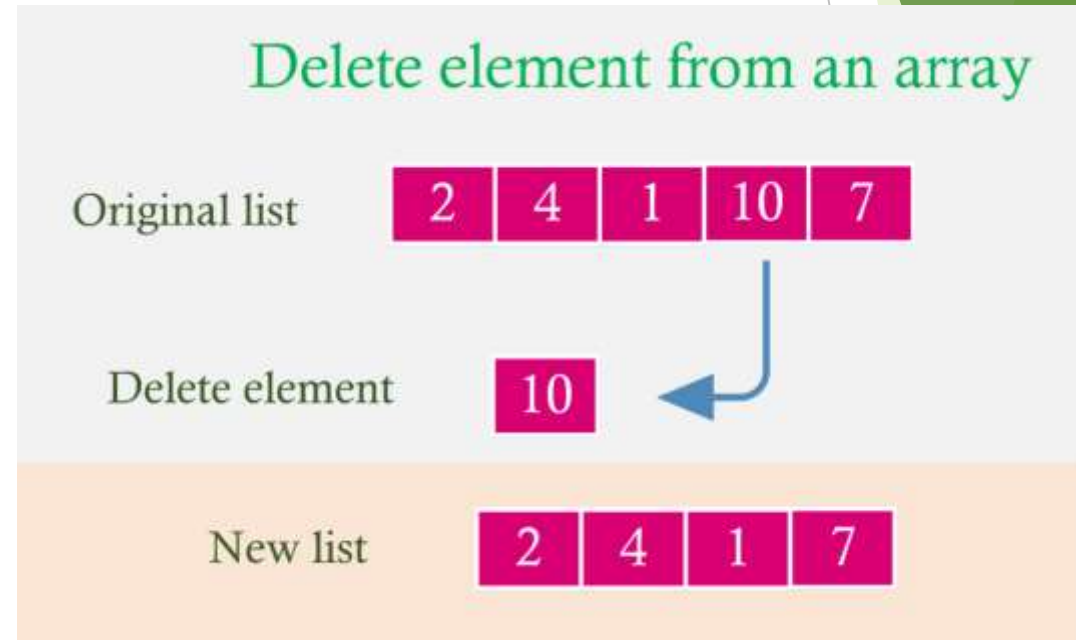
The array elements after deletion :

$LA[0] = 2$

$LA[1] = 4$

$LA[2] = 1$

$LA[3] = 7$





Searching Operation

- 1 Start
2. Set $J = 0$
3. Repeat steps 4 and 5 while $J < N$
4. IF $LA[J]$ is equal ITEM THEN GOTO STEP 6
5. Set $J = J + 1$
6. PRINT J, ITEM
7. Stop

The original array elements are :

$LA[0] = 20$

$LA[1] = 40$

$LA[2] = 10$

$LA[3] = 30$

$LA[4] = 60$

Found element 30 at position 3

Sequential Search

Index:	0	1	2	3	4
Value:	20	40	10	30	60

Target = 30

Step 1: Compare 30 with value at index 0

Step 2: Compare 30 with value at index 1

Step 3: Compare 30 with value at index 2

Step 4: Compare 30 with value at index 3 (success)



Update Operation

1. Start
2. Set $LA[K-1] = \text{ITEM}$
3. Stop

The original array elements are :

$LA[0] = 1$

$LA[1] = 3$

$LA[2] = 5$

$LA[3] = 7$

$LA[4] = 8$

The array elements after updation :

$LA[0] = 1$

$LA[1] = 3$

$LA[2] = 10$

$LA[3] = 7$

$LA[4] = 8$



Assessment

Usually, the index for the first element of an array is _____?

- a) 0
- b) 1
- c) 2
- d) -1

What is right way to Initialize array?

- A. `int num[6] = { 2, 4, 12, 5, 45, 5 };`
- B. `int n{ } = { 2, 4, 12, 5, 45, 5 };`
- C. `int n{6} = { 2, 4, 12 };`
- D. `int n(6) = { 2, 4, 12, 5, 45, 5 };`

What will be the output of this following program?

```
main()
{
    printf("I MCA");
    main();
}
```

- A. It will keep on Print I MCA once
- B. It will Print I MCA once
- C. Wrong statement
- D. None of the these



References

1. Tanaenbaum A.S., Langram Y. Augestein M.J “Data Structures using C”, Pearson Education , 2008.
2. <https://www.tutorialpoint.com>
3. https://www.youtube.com/watch?v=551-aZ7_F24

