

Data Structures and Algorithms

Lecture 3: Arrays – Traversal, Insertion and Deletion

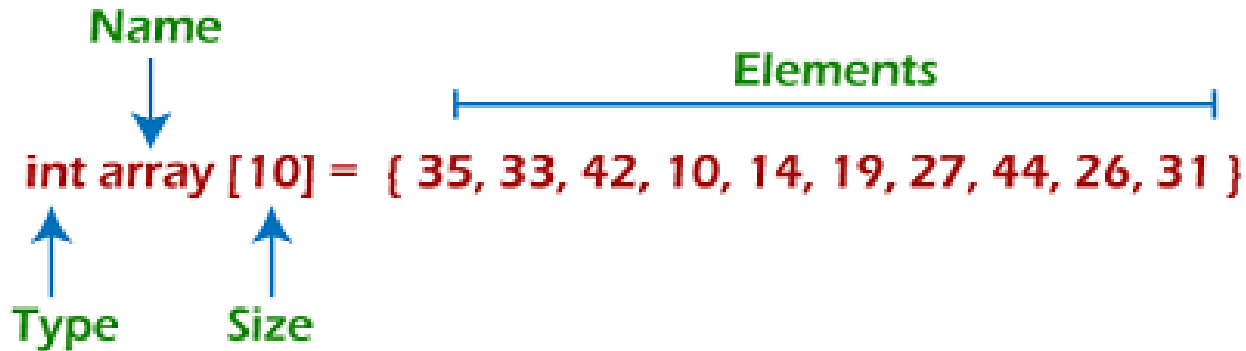
Arrays

Arrays are defined as the collection of similar types of data items stored at contiguous memory locations.

Properties of array:

- Each element in an array is of the same data type and carries the same size.
- Elements in the array are stored at contiguous memory locations from which the first element is stored at the smallest memory location.
- Elements of the array can be randomly accessed since we can calculate the address of each element of the array with the given base address and the size of the data element.

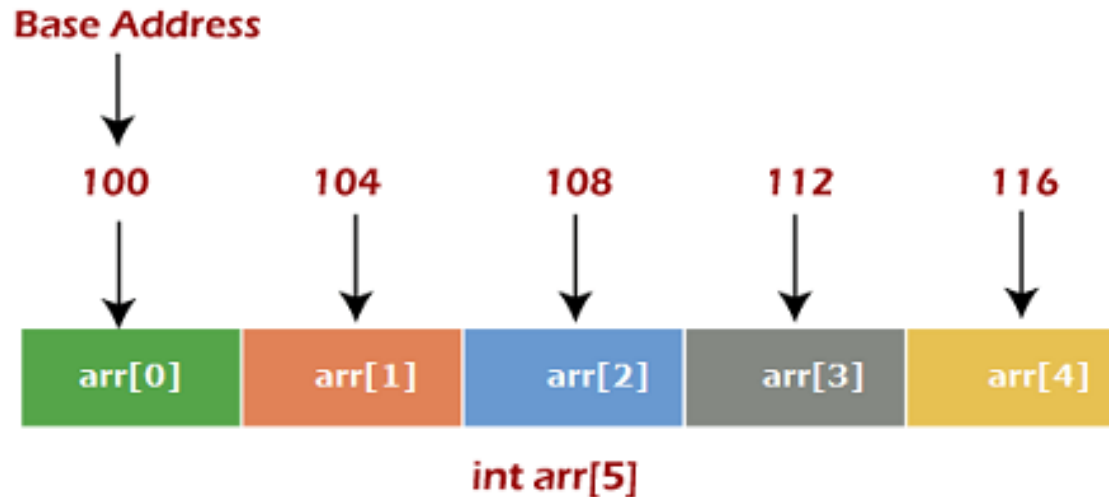
Representation of an Array



As per the above illustration, there are some of the following important points -

- Index starts with 0.
- The array's length is 10, which means we can store 10 elements.
- Each element in the array can be accessed via its index.

Memory Allocation of an Array



1. The name of the array represents the base address or the address of the first element in the main memory.
2. Each element of the array is represented by proper indexing.

Access an Element from the Array

The below possible are the ways to access the array elements:

1. Base Address of the array.
2. Size of an element in bytes.
3. Type of indexing, array follows.

Types of Arrays

One dimensional array:

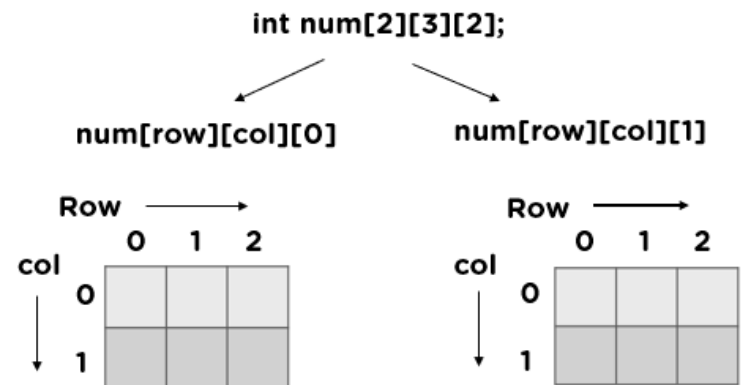
Elements are stored one after another.



Multi-dimensional array:

- Two-dimensional array: like a table where each cell contains elements.
- Three-dimensional array: like a cuboid made up of smaller cuboids where each cuboid can contain an element.

Col	0	1	2
Row 0	1	2	3
1	4	5	6
2	7	8	9



Basic operations in an Array

The basic operations supported in the array -

1. Traversal - This operation is used to print the elements of the array.
2. Insertion - It is used to add an element at a particular index.
3. Deletion - It is used to delete an element from a particular index.
4. Search - It is used to search an element using the given index or by the value.
5. Update - It updates an element at a particular index.

Traversal Operation in an Array

Traversal in an array is a process of visiting each element once.



Traversal Operation in an Array

```
#include<stdio.h>

int main()
{
    int a[5] = {2, 3, 5, 7, 11};
    for(int i=0;i<5;i++)
    {
        //traversing ith element in the array
        printf("%d\n",a[i]);
    }
    return 0;
}
```

Output:

2
3
5
7
11

Insertion Operation in an Array

Insertion in an array is the process of including one or more elements in an array.

Insertion of an element can be done:

At the beginning

At the end and

At any given index of an array.



Insertion Operation in an Array

At the Beginning:

```
#include<stdio.h>
int main()
{
    int array[10], n,i, item;
    printf("Enter the size of array: ");
    scanf("%d", &n);
    printf("\nEnter Elements in array: ");
    for(i=0;i<n;i++)
    {
        scanf("%d", &array[i]);
    }
    printf("\n Enter the element at the beginning:");
    scanf("%d", &item);
    n++;
```

```
for(i=n; i>1; i--)
{
    array[i-1]=array[i-2];
}
array[0]=item;
printf("Resultant array element:");
for(i=0;i<n;i++)
{
    printf("%d ", array[i]);
}
getch();
return 0;
}
```

Enter the size of array: 5
Enter elements in array: 2 3 5 7 11
Enter the element at the beginning: 1
Resultant array element:
1 2 3 5 7 11

Insertion Operation in an Array

At the End:

```
#include<stdio.h>
#include<conio.h>
int main() {
    int array[10], i, values;
    printf("Enter 5 Array Elements: ");
    for(i=0; i<5; i++)
        scanf("%d", &array[i]);
    printf("\nEnter element to insert: ");
    scanf("%d", &values);
    array[i] = values;
    printf("\nThe New Array is:");
    for(i=0; i<6; i++)
        printf("%d ", array[i]);
    getch();
    return 0;
}
```

Enter 5 Array Elements: 2 3 5 7 11
Enter element to insert: 13
The New Array is: 1 2 3 5 7 11 13

Insertion Operation in an Array

At a Specific Position:

```
#include <stdio.h>
int main()
{
    int array[100], pos, size, val;
    printf("Enter size of the array:");
    scanf("%d", &size);
    printf("\nEnter %d elements:", size);
    for (int i = 0; i < size; i++)
        scanf("%d", &array[i]);
    printf("\nEnter the insertion location:");
```

```
scanf("%d", &pos);
printf("\nEnter the value to insert:");
scanf("%d", &val);
for (int i = size - 1; i >= pos - 1; i--)
    array[i+1] = array[i];
array[pos-1] = val;
printf("\nResultant array is:");
for (int i = 0; i <= size; i++)
    printf("%d\n", array[i]);
return 0;
}
```

Enter size of the array: 5

Enter 5 elements: 2 3 5 7 11

Enter the insertion location: 2

Enter the value to insert: 13

Resultant array is: 2 13 3 5 7 11

Deletion Operation in an Array

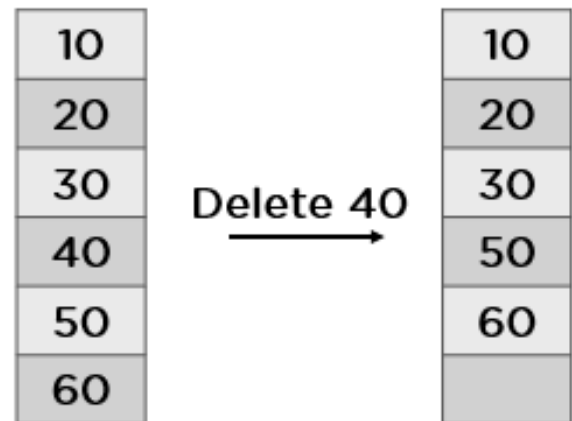
Deletion of an element is the process of removing the desired element and re-organizing it.

Deletion of an element can be done:

At the beginning

At the end

Before deletion After deletion



Deletion Operation in an Array

At the Beginning:

```
#include<stdio.h>

int main()
{
    int n,array[10];
    printf("enter the size of an array");
    scanf("%d" ,&n);
    printf("\nenter elements in an array");
    for(int i=0;i<n;i++)
        scanf("%d", &array[i]);
    n--;
```

```
    for(int i=0;i<n;i++)
        array[i]=array[i+1];
    printf("\nafter deletion ");
    for(int i=0;i<n;i++)
        printf("\n%d" , array[i]);
}
```

enter the size of an array 5
enter elements in an array 2 3 5 7 11
after deletion 3 5 7 11

Deletion Operation in an Array

At the End:

```
#include<stdio.h>

int main() {
    int n,array[10];
    printf("enter the size of an array");
    scanf("%d" ,&n);
    printf("enter elements in an array");
    for(int i=0;i<n;i++)
        scanf("%d", &array[i]);
    printf("\nafter deletion");
    for(int i=0;i<n-1;i++)
        printf("\n%d" , array[i]);
}
```

enter the size of an array 5
enter elements in an array 2 3 5 7 11
after deletion 2 3 5 7

Recall:

- Array and properties of array
- Representation of an Array
- Memory allocation of an array
- Accessing elements of an array
- Types of array
- Array Operations – Traversal, Insertion and Deletion

