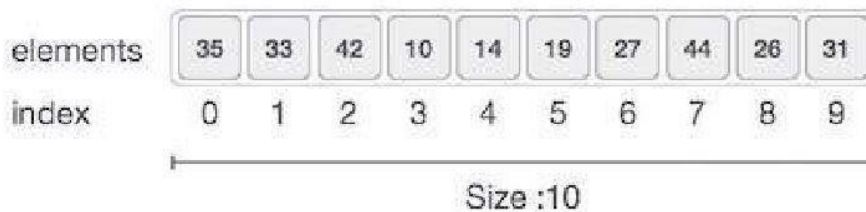


Lecture 2 - Arrays

Array is a container which can hold a fix number of items and these items should be of the same type. Most of the data structures make use of arrays to implement their algorithms.

Array Representation

Arrays can be declared in various ways in different languages. For illustration, let's take C array declaration.



As per the above illustration, following are the important points to be considered.

- Index starts with 0.
- Array length is 10 which means it can store 10 elements.
- Each element can be accessed via its index. For example, we can fetch an element at index 6 as 19.

Basic Operations Following are the basic operations supported by an array.

- Traverse – Prints all the array elements one by one.
- Insertion – Adds an element at the given index.
- Deletion – Deletes an element at the given index.
- Search – Searches an element using the given index or by the value.
- Update – Updates an element at the given index.

Insertion Operation

Insert operation is to insert one or more data elements into an array. Based on the requirement, a new element can be added at the beginning, end, or any given index of array.

Example:

Let **LA** be a Linear Array (unordered) with **N** elements and **K** is a positive integer such that **$K \leq N$** . Following is the algorithm where ITEM is inserted into the Kth position of LA –

1. Start
2. Set $J=N$
3. Set $N = N+1$
4. Repeat steps 5 and 6 while $J \geq K$
5. Set $LA[J+1] = LA[J]$
6. Set $J = J-1$
7. Set $LA[K] = \text{ITEM}$
8. Stop

Deletion Operation

Deletion refers to removing an existing element from the array and re-organizing all elements of an array.

Algorithm Consider LA is a linear array with N elements and K is a positive integer such that $K \leq N$. Following is the algorithm to delete an element available at the Kth position of LA.

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

Search Operation

You can perform a search for an array element based on its value or its index.

Algorithm Consider **LA** is a linear array with **N** elements and **K** is a positive integer such that $K \leq N$. Following is the algorithm to find an element with a value of ITEM using sequential search

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

Stop

LA=[10 52 5 87 96 5 3]

ITEM=96

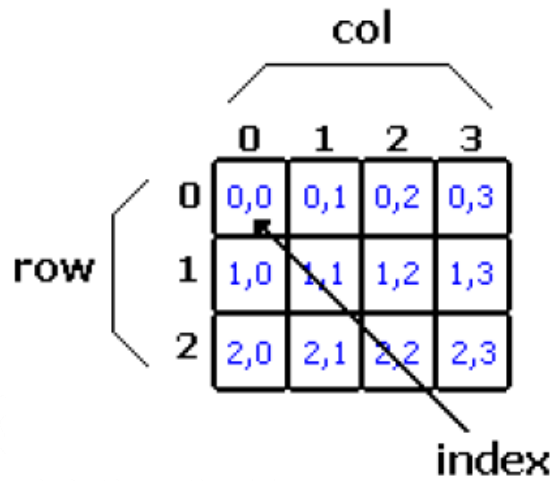
Array of Two Dimensions

Arrays can have higher dimension. There can be arrays of two dimension which is array of arrays. It is accessed with two index. Also there can be arrays of dimension higher than two.

Examples:

A: array[1..3,1..4] of integer

```
int [,] num = new int [3,4];
```



Initializing 2D-Array Elements

```
int [,] a = new int [2,3] = { { 1, 2, 3 } , { 4, 5, 6 } };
```



Example:

Write C++ program, to read 4*4 2D-array, then find the **summation** of the array elements.

```
namespace ConsoleApplication3
{
    class Program
    {
        static void Main(string[] args)
        {int [,] a=new int[4,4];
            int i ,j, sum = 0;
            for ( i = 0 ; i < 4; i++ )
            for ( j = 0 ; j < 4; j++ )
                a[i,j]=int.Parse(Console.ReadLine());
            for ( i = 0 ; i < 4; i++ )
            for ( j = 0 ; j < 4; j++ )
                sum += a [i,j];
            Console.WriteLine("summation is: ",sum );
            for ( i = 0 ; i < 4; i++ )
            { for ( j = 0 ; j < 4; j++ )
                Console.WriteLine( a[i,j]);
            }
        }
    }
}
```