3.1 Introduction:

Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type.

An array is used to store a collection of data, but it
is often more useful to think of an array as a collection of variables of the same type.

Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers[] and use
numbers[0], numbers[1], and ..., numbers[99] to represent individual variables.

A specific element in an array is accessed by an index.

All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.

- **Array :**
- **Definition:**An array is a collection of homogenous data stored under unique name.
- values in an array are called as 'elements of an array'. The first index in the array is numbered 0, so that the last index is 1 less than the size of array.
  1. An array is also known as a subscripted variable.
  2. Before using an array its type & dimensions must be declared.
  3. However big an array its elements are always stored in contagious memory locations.

Index (x)	0	1	2	3	4
Value a[x]	12	12	45	15	1
Location	2002	2004	2006	2008	2010

e.g. x is a 5 element array i.e. a[5]

First statement is referred to as a[0] and last element is referred to as a[4]

i.e. for an n-element array the subscripts always range from 0 to n – 1

- **Need of Array:** Suppose we want to process marks of 50 students then one approach is that define 50 variables which is tedious. Another approach is that just define a single variable which can store 50 values i.e. Array.

Types of an Array

1. One / Single Dimensional Array
2. Two Dimensional Array

3.2 One Dimensional Array:

The array which is used to represent and store data in a linear form is called as 'single or one dimensional array.

- **Defining array/ Array Declaration :**

  Syntax: Data_type array_name [no. of values]

  We can define array as a normal variable as shown below

  int a[5];   /* to define array of 5 integers */

  char c[10]; /* to define array of 10 characters */

  Array may be single dimension as we have defined above or multi-dimensional as below.

  int a[3][3]; /* to define array of 3 X 3 matrix of integers */

  int c[3][3][3]; /* to define array of 3 X 3 X 3 matrix of integers */

Initializing Arrays

You can initialize an array in C either one by one or using a single statement as
follows:

double balance[5] = {1000.0, 2.0, 3.4, 7.0, 50.0};

The number of values between braces { } cannot be larger than the number ofelements that we declare for the array between square brackets [ ].

If you omit the size of the array, an array just big enough to hold the initialization is created. Therefore, if you write:

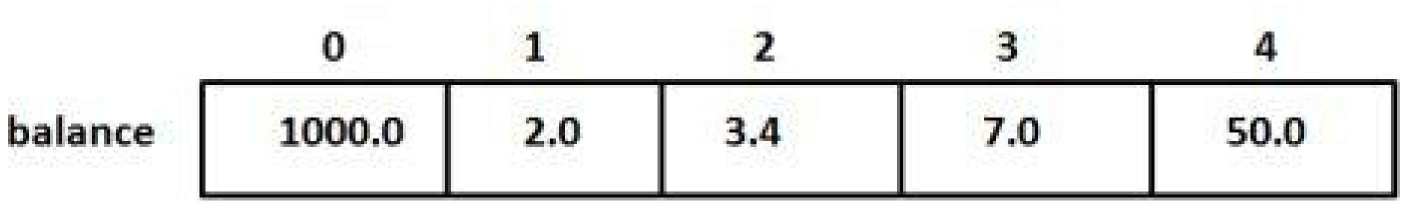
double balance[] = {1000.0, 2.0, 3.4, 7.0, 50.0};

You will create exactly the same array as you did in the previous example.

Following is an example to assign a single element of the array:

balance[4] = 50.0;

The above statement assigns the 5th element in the array with a value of 50.0.
All arrays have 0 as the index of their first element which is also called the base index and the last index of an array will be total size of the array minus 1.
Shown below is the pictorial representation of the array we discussed above:



Examples:

int a[3] = {2, 3, 5};

char ch[20] = "Deesha" ;

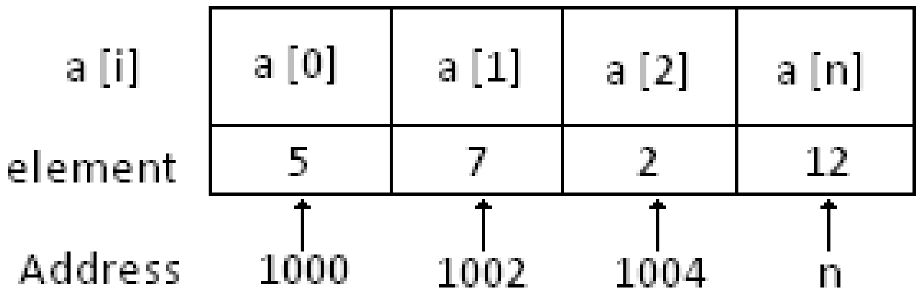
float stax[3] = {5003.23, 1940.32, 123.20} ;

Total Size (in Bytes):

total size = length of array * size of data type

In above example, a is an array of type integer which has storage size of 3 elements. The total size would be 3 * 2 = 6 bytes.

* Memory Allocation :



Memory allocation for one dimensional array

Accessing Array Elements

An element is accessed by indexing the array name. This is done by placing the index of the element within square brackets after the name of the array. For example:

double salary = balance[9];

The above statement will take the 10th element from the array and assign the value to salary variable.

Note the following points carefully:

1. Till the array elements are not given any specific values, they are supposed to contain garbage values.
2. If the array is initialized where it is declared, mentioning the dimension of the array is optional

The following example shows how to use all the three above-mentioned concepts viz. :

```
#include <stdio.h>
int main ()
{
int n[ 10 ]; /* n is an array of 10 integers */
inti,j;
/* initialize elements of array n to 0 */
for ( i = 0; i < 10; i++ )
{
n[ i ] = i + 100; /* set element at location i to i + 100 */
}
/* output each array element's value */
for (j = 0; j < 10; j++ )
{
printf("Element[%d] = %d\n", j, n[j] );
}
return 0;
}
```

When the above code is compiled and executed, it produces the following result:

```
Element[0] = 100
Element[1] = 101
Element[2] = 102
Element[3] = 103
```

```
Element[4] = 104
Element[5] = 105
Element[6] = 106
Element[7] = 107
Element[8] = 108
        Element[9] = 109
```

e.g. w**rite a program to define input, output an array.**

```
        void main( )
        {       int a[10], i;
                clrscr();
                for( i = 0; i <= 5;  i++)
                {       printf(“enter %d element :”, i+1);
                        scanf(“%d”, &a[i]);
                }
        printf(“ 3rd Element is %d \n”,a[2]);
        printf(“ 2rd Element is %d \n”,a[1]);

    getch();
        }
```

e.g. **Write a program for addition & subtraction of one dimensional matrix**

```
        void main()
        {           int a[5],b[5],i,s=0,c[5],d[5],s1=0;
                clrscr();
                printf("enter 5 nos. of a[i]=\n");
                for(i=0;i<5;i++)
        scanf("%d" ,&a[i]);
        printf("enter 5 nos. of b[i]=\n");
        for(i=0;i<5;i++)
                scanf("%d" ,&b[i]);
        for(i=0;i<5;i++)
                c[i]=a[i]+b[i];
        printf("TOTAL SUM=\n");
        for(i=0;i<5;i++)
                printf("%d\n" ,c[i]);
        printf("TOTAL SUBTRACTION=");
        for(i=0;i<5;i++)
                d[i]=a[i]-b[i];
        for(i=0;i<5;i++)
                printf("%d\n" ,d[i]);
        getch();
}
```

3.3 Two Dimensional Array:- Suppose a college has 2 divisions of 60 students each then representation is like int students[120]; & is not matching real life instead int students[2][60];matches with real life.

NOTE:-

Visualizing multidimensional arrays ‘C’ provides arrays up to any dimensions i.e. 2D,3D …. & So on. A 2D array is an array in which element is 1D array.

e.g. int x[2][3]; This array contains only two elements each of which is 1D array.

Similarly 3D array is an array, In which each element is 2D array.

Thus nD array is an array in which each element is n-1D array.

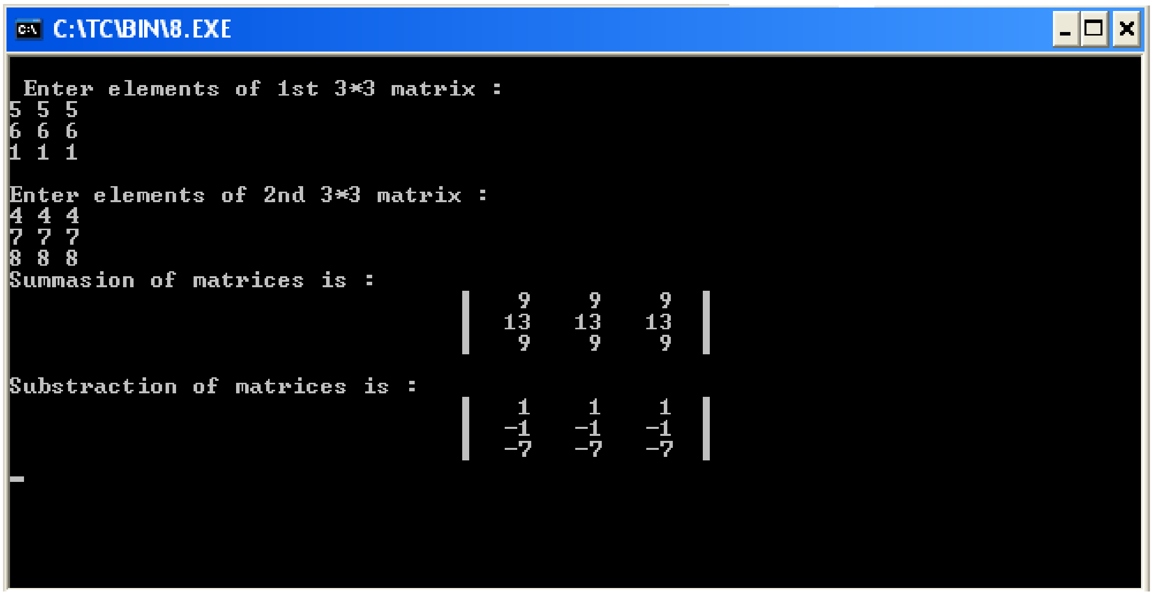
e.g. **write a program for addition & subtraction of 3 x 3 matrix.**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[3][3],b[3][3],i,j,c[3][3];
clrscr();
printf("\n Enter elements of 1st 3*3 matrix :\n");

for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
scanf("%d",&a[i][j]);
}
printf("\nEnter elements of 2nd 3*3 matrix :\n");

for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
scanf("%d",&b[i][j]);
```

```
}
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
c[i][j]=a[i][j]+b[i][j];
}
printf("Summasion of matrices is :\n");

for(i=0;i<3;i++)
  {
      printf("\t\t\t\tÝ ");
      for(j=0;j<3;j++)
      printf("%3d  ",c[i][j]);
      printf("Ý \n");
  }

for(i=0;i<3;i++)
  {
      for(j=0;j<3;j++)
      c[i][j]=a[i][j]-b[i][j];
  }



      printf("\nSubstraction of matrices is :\n");
      for(i=0;i<3;i++)
            {
                  printf("\t\t\t\tÝ ");
```

```
                    for(j=0;j<3;j++)
                    printf("%3d  ",c[i][j]);
                    printf("Ý\n");
               }
       getch();
}
```

Output:-



C:\TC\BIN\8.EXE

```
 Enter elements of 1st 3*3 matrix :
5 5 5
6 6 6
1 1 1

Enter elements of 2nd 3*3 matrix :
4 4 4
7 7 7
8 8 8
Summasion of matrices is :
                                 |   9    9    9  |
                                 |  13   13   13  |
                                 |   9    9    9  |

Substraction of matrices is :
                                 |   1    1    1  |
                                 |  -1   -1   -1  |
                                 |  -7   -7   -7  |

_
```

Ex:1

WAP to input an Array of 5 elements and display the sum of all values

```
 #include<stdio.h>
#include<conio.h>
void main()
{
int arr[5],sum=0,i;
clrscr();
printf("\n\t Enter 5 Array values :");
for(i=0;i<5;i++)
{
scanf("%d",&arr[i]);
}
for(i=0;i<5;i++)
{
sum = sum + arr [i];
}
printf("\n\t Addition = %d",sum);
getch();
}
```

Ex:2

WAP to input marks of 10 students and display the sum ,average and variance of marks.