

Array

Single Dimensional Array: Single Dimensional Array is a data structure which is used to store homogeneous (same type) of data elements. The elements of an array are stored in a contiguous memory location.

Syntax of Declaration of Single Dimensional Array:

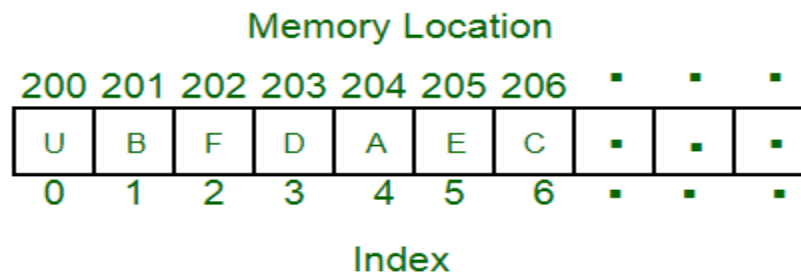
type Arrayname [size];

Example:

int A[10];

char B[5];

float C[6];



Advantages of Array

1. An array is a better way of storing data of same type.
2. An element in the array can be directly accessed by using the index of an array.
3. Traversing of an array is very easy.

Disadvantages of Array

1. Fixed size of array means it allows to enter only fixed number of elements.
2. Insertion and Deletion from the array would be costly.

Representation of Single Dimensional Array:

int a[5]={ 10,15,1,3,20};

a[0]	a[1]	a[2]	a[3]	a[4]
10	15	1	3	20
1000	1002	1004	1006	1008

Base Address: the address of first element in an array is called as Base address.

Index Number: If the size of an Array is 'n' then the indexing is starts from 0 to n-1.

Length: Length of an Array is equal to the size of an array.

Address Calculation in a Single Dimension Array:

Address of an element of a single dimensional array is calculated by using the following formula:

$$\text{Address of } A[I] = B + W * (I - L_b)$$

Where,

B= (Base Address): The address of first element in an array.

W= (width): storage size (in bytes) acquired by one element in array.

I= subscript of element whose address is to be found.

L_b = Lower Bound.

$$\text{Size of Array} = U_b - L_b + 1$$

Where,

U_b = UpperBound

L_b = Lower Bound

Total memory size = Size of Array * word size

Example 1:

Given the base address of an array A [1300-----1900] as 1020 and size of each element is 2 bytes in memory. Find the address of A [1700].

Solution:

Address of A[I] = $B + W * (I - LB)$

Address of A [1700] = $1020 + 2 * (1700 - 1300)$

Address of A [1700] = $1020 + 2 * (400)$

Address of A [1700] = $1020 + 800$

Address of A [1700] = 1820