**ARRAY**

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.

Thus **array** ia a collection of homogeneous pieces of data that are all identical in type and stored in consecutive memory locations.for example,**A ia an integer array** storing 10 integer numbers.

Index->0 1 2 3 4 5 6 7 8 9

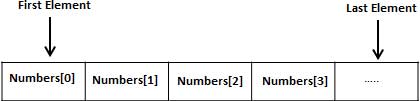
A 20 30 40 56 67 89 12 43 65 32

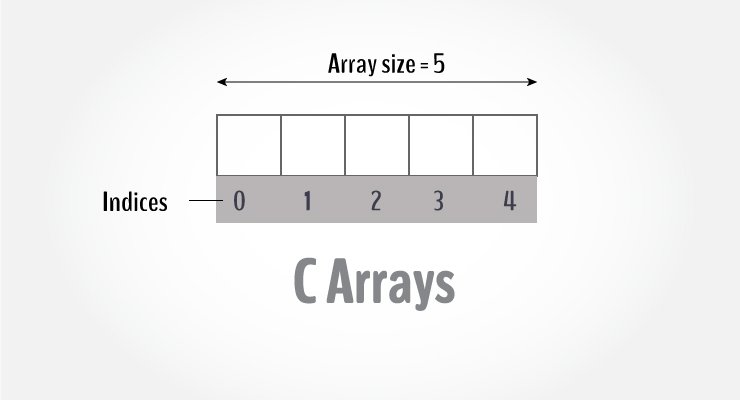
**Fig:example of integer array storing 10 integers**.

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

Now,let us assume that the base address of A is 5012.Base address of an array is the memory address of the first element in the array.So here 5012 is the memory address of 20.Next the memory address of the second element of A is 5014 as the memory size of int type variable is 2 bytes.In this way,the memory address of the third and fourth element of A is 5016 and 5018 respectively.

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.





An array is a variable that can store multiple values. For example, if you want to store 100 integers, you can create an array for it.

int data[100];

TYPES OF ARRAY

1)**one –dimensional array**:In one dimensional array,data is stored row or column wise and hold in consecutive memory locations.

2)**multi-dimensional array**:In multi-dimensional array,data is held both row and column wise.The declaration syntax of multidimensional array is **datatype arrayname [size 1] [size 2] [size 3]..............[size N].**For example declaration of a 2 dimensional array is **float arrtwo[30][20];**A 2-D array is also called as a matrix

How to declare an one dimensional array?

To declare an array in C, a programmer specifies the type of the elements and the number of elements required by an array as follows −

type arrayName [ arraySize ];

## This is called a *single-dimensional* array. The arraySize must be an integer constant greater than zero and type can be any valid C data type

dataType arrayName[arraySize];

**For example,**

float mark[5];

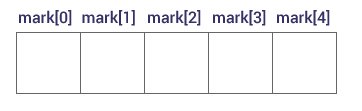
Here, we declared an array, mark, of floating-point type. And its size is 5. Meaning, it can hold 5 floating-point values.

It's important to note that the size and type of an array cannot be changed once it is declared.

**Access Array Elements**

You can access elements of an array by indices.

Suppose you declared an array mark as above. The first element is mark[0], the second element is mark[1] and so on.



Declare an Array

#### ****Few keynotes****:

* Arrays have 0 as the first index, not 1. In this example, mark[0] is the first element.
* If the size of an array is n, to access the last element, the n-1 index is used. In this example, mark[4]
* Suppose the starting address of mark[0] is **2120d**. Then, the address of the mark[1] will be **2124d**. Similarly, the address of mark[2] will be **2128d** and so on.  
  This is because the size of a float is 4 bytes.

**How to initialize an array?**

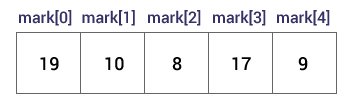
It is possible to initialize an array during declaration. For example,

int mark[5] = {19, 10, 8, 17, 9};

You can also initialize an array like this.

int mark[] = {19, 10, 8, 17, 9};

Here, we haven't specified the size. However, the compiler knows its size is 5 as we are initializing it with 5 elements.



Initialize an Array

Here,

mark[0] is equal to 19

mark[1] is equal to 10

mark[2] is equal to 8

mark[3] is equal to 17

mark[4] is equal to 9

**character arrays** may be initialized in a similar manner.for example

char name[]={‘J’,’O’,’H’,’N’,’\0’};

declares the name to be an array of five characters ,initialized with the string “JOHN” ending with the **null character**.

Or char name[]=”JOHN”;

For example1 : Char city[9]=”NEW YORK”;

Char city[9]={‘N’,’E’,’W’,’ ’,’Y’,’O’,’R’,’K’,’\0’};

For example 2: Char b[9]=”COMPUTER”; //correct

Char b[8]=”COMPUTER”; //wrong

**NOTE**: A null character(‘\0’) is assigned at end of the string.

**Example 1: Array Input/Output**

// Program to take 5 values from the user and store them in an array

// Print the elements stored in the array

#include <stdio.h>

int main()

{

int values[5]; //array declaration

printf("Enter 5 integers: ");

// taking input and storing it in an array

for(int i = 0; i < 5; ++i) {

scanf("%d", &values[i]);

}

printf("Displaying integers: ");

// printing elements of an array

for(int i = 0; i < 5; ++i) {

printf("%d\n", values[i]);

}

return 0;

}

**Output**

Enter 5 integers: 1

-3

34

0

3

Displaying integers: 1

-3

34

0

3

Here, we have used a for loop to take 5 inputs from the user and store them in an array. Then, using another for loop, these elements are displayed on the screen

**Example 2: Calculate Average**

// Program to find the average of n numbers using arrays

#include<stdio.h>

int main()

{

int marks[10], i, n, sum = 0, average;

printf("Enter number of elements: ");

scanf("%d",&n);

for(i=0; i<n; ++i)

{

printf("Enter number%d: ",i+1);

scanf("%d",&marks[i]);

// adding integers entered by the user to the sum variable

sum += marks[i];

}

average = sum/n;

printf("Average = %d", average);

return 0;

}

**Output**

Enter n: 5

Enter number1: 45

Enter number2: 35

Enter number3: 38

Enter number4: 31

Enter number5: 49

Average =39