Array Basics

An array is a collection of data items of the same data type. And it is also known as a subscript variable.

- Items are stored at contiguous memory locations in arrays.
- It can also store the collection of derived data types such as pointers, structures, etc.
- The C Language places no limits on the number of dimensions in an array. This means we can create arrays of any number of dimensions. It could be a 2D array or a 3D array or more.

Dimensions of an array

- A one-dimensional array is like a list.
- A two-dimensional array is like a table.

Some texts refer to one-dimensional arrays as vectors and two-dimensional arrays as matrices and use the general term arrays when the number of dimensions is unspecified or unimportant.

How do Arrays help?

Programs that use arrays for managing a large number of the same data type variables are more organized and readable in comparison to creating different variables of the same type for each data element.

Arrays allow us to create many variables by just a single line. It means there is no need to create or specify every variable.

Advantages of Arrays?

- It is used to represent multiple data items of the same type by using only a single name.
- Accessing any random item at any random position in a given array is very fast in an array.
- There is no case of memory shortage or overflow in the case of arrays since the size is fixed and elements are stored in contiguous memory locations.

Properties of Arrays

- An array stores data be it of any data type in contiguous memory locations in RAM.

- Each element of an array is of the same size i.e. their data types are the same so the memory consumed by each is also the same.
- Any element of the array with a given index can be accessed very quickly by using its address which can be calculated using the base address and the index number.
- **Index Number** – It is the special type of number which allows us to access variables of arrays. Index number provides a method to access each element of an array in a program.

Array Operations

1. Defining an Array

a) Without specifying the size

```
int arr[] = {1, 2, 3};
```

- Square brackets can be left empty.
- Array must have elements; it cannot be empty.

b) With specifying the size

```
int arr[3];

arr[0] = 1;

arr[1] = 2;

arr[2] = 3;
```

- Size is defined during declaration.
- Elements can be assigned later.

---

2. Accessing an Array Element

- Array elements are accessed using their **index**.
- Indexing starts from **0**.

Example:

```
#include <stdio.h>


int main() {
    int arr[] = {1, 5, 7, 2};
    printf("%d ", arr[2]); // printing element at index 2
}
```

Output:

```
7
```

---

3. Changing an Array Element

- Array elements can be **overwritten** using their index.

Example:

```
#include <stdio.h>

int main() {
    int arr[] = {1, 5, 7, 2};
    arr[2] = 8; // changing the element at index 2
    printf("%d ", arr[2]); // printing element at index 2
}
```

Output:

```
8
```