Pointer to an array

“We can point the whole array using pointers. Using the array pointer, we can easily manipulate the multi-dimensional array.”

**Example:**

**char** arr[**5**] = {**‘a’**, **‘b’**, **‘c’**, **‘d’**, **‘e’**};

**char** (\*ptr)[**5**];

ptr = &arr;

ptr points to whole array instead of only one element of the array.

We use parenthesis to pronounce pointer to an array. Since subscript has higher priority than indirection, it is crucial to encase the indirection operator and pointer name inside brackets.

**Pointer to Multidimensional Array**

Let's see how to make a pointer point to a multidimensional array. In **a[i][j],**a will give the base address of this array, even a + 0 + 0 will also give the base address, that is the address of **a[0][0]** element.

**\*(\*(a + i) + j)**

Array Of Pointers

“Is an array that consists of variables of pointer type, which means that the variable is a pointer addressing to some other element.”

**Example:**

int \*var\_name[Size];

int \*ptr[5];

As we know, arrays are collections of elements stored in contiguous memory locations. An array of pointers is similar to any other array in C Language. It is an array which contains numerous pointer variables and these pointer variables can store address values of some other variables having the same data type.

An application of an array of pointers is that it becomes easy to store strings in a char pointer array and it also reduces the memory consumption by the dynamic Allocation in the heap.

