# **1D, 2D, MultiDimensional Array Assignments**

**Mandatory**

**1D Array**

1. **Refer the code snippet and answer the queries**

**int main()**

**{**

**int array[100];**

**int \*ptr;**

**// do something**

**}**

**Q1: Can pointer be used in Array-style syntax? e.g. ptr[10], ptr[0]**

Yes, a pointer can be used in array-style syntax. This is because ptr[10] is equivalent to \*(ptr + 10) and ptr[0] is equivalent to \*ptr.

**Q2: Can Array be used in Pointer-style syntax? e.g. \*array, \*(array + 0), \*(array + 10)**

Yes, arrays can be used in pointer-style syntax. An array name in C is a constant pointer to the first element of the array.

**Q3: is ptr++ valid?**

Yes, ptr++ is valid. It increments the pointer, making it point to the next element in memory(it moves to the next address corresponding to the type of the pointer)

**Q4: is array++ valid?**

No, array++ is not valid. In C, array is not a modifiable lvalue because it is a constant pointer to the first element of the array. Arrays cannot be incremented as pointers can.

**Q5: what is sizeof(array)?**

Sizeof(array) gives the total size of array

**Q6: what is sizeof(ptr)?**

Gives the sizeof ptr. In most systems, generally 4 bytes.

1. **Refer the code snippet below. Comment on the other elements (other than those that are explicitly initialized) of all array variables in code snippet below.**

**#define MAX 100**

**int main()**

**{**

**int arr[MAX] = {11,22,33};**

**int arr1[MAX]={0};**

**static int arr2[MAX];**

**}**

**arr[MAX] = {11, 22, 33};**

* The first three elements (arr[0], arr[1], arr[2]) are explicitly initialized to 11, 22, and 33, respectively. All the other elements (arr[3] to arr[99]) will be initialized to 0.

**arr1[MAX] = {0};**

* All elements of arr1 will be initialized to 0.

**static int arr2[MAX];**

* All elements of arr2 will be initialized to 0 as they are static variables.

1. **Refer the program “array\_pointer.c”. Add a function getmax() to find the maximum in the array and call in main() and display the result.**

**A computer screen shot of a black screen with colorful text

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

1. **Extend the code given below to read N and a start value from the user to perform the given operations.**

**#define MAX 100**

**int main()**

**{**

**int arr[MAX] = {11,22,33};**

**}**

**Add the following functions choosing proper input, output and return.**

1. **init() - Use the inputs to initialize the first N elements of the array with N consequetive values starting with given start value .**
2. **update() – increment value of every element in the array**
3. **display() – display the contents of array**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

**2D, MultiDimensional Arrays**

1. **Implement sort() to sort a given array. Refer the code snippet below.**

**int main()**

**{**

**char arr[]= “xaybz”;**

**sort(arr, sizeof(arr)/sizeof(arr[0]);**

**return 0;**

**}**

**A screenshot of a computer code

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

1. **Refer the code snippet below.**

**int main()**

**{**

**char arr[][3] = {**

**sort(arr, sizeof(arr)/sizeof(arr[0]);**

**return 0;**

**}**

**Allow user to perform the following operations.**

* 1. **init() - initialize the array and return 0**
  2. **search\_update() – search for a given element in array and if found update it to given value and return 0 else return 1**
  3. **display() – traverse and display array contents**

**For the functions, pass array and other required arguments to functions and return as per requirement**

**A screenshot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**