# 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]**

Ans: Yes, pointer can be used in Array-style syntax. For eg ptr[10], ptr[0], will return the base address of that element base address when declared.

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

Ans: Yes, array can be used in pointer-style syntax as it will print the value of that accessed element.

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

Ans: Yes, it is valid because would dereference the pointer and after that increment the pointer by the size of the memory type it is pointing to.

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

Ans: No, it is no valid.

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

Ans: sizeof(arr) is the total size occupied by the array.

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

Ans: sizeof(ptr) refers to the size of pointer by referring to the value of address of that particular element but it is not fixed as it depends on other factors.

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];

}

Ans:

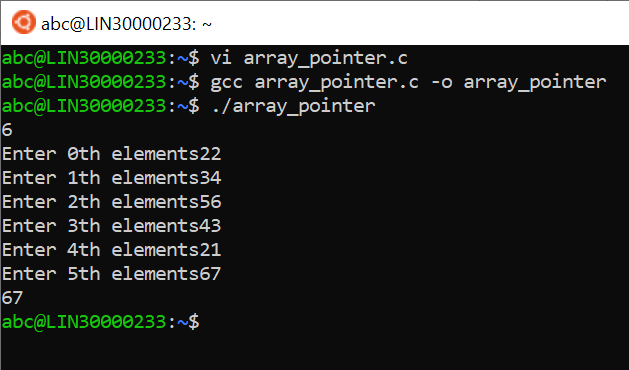
int arr[MAX] = {11,22,33}; //assigning to first 3 elements

int arr1[MAX] = {0}; //return value 0

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.**

Ans:





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()

{

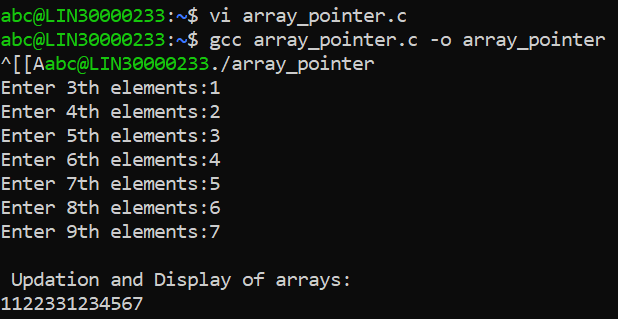
int arr[MAX] = {11,22,33};

}

**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**

Ans: 



**2D, MultiDimensional Arrays**

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

int main()

{

char arr[]= “xaybz”;

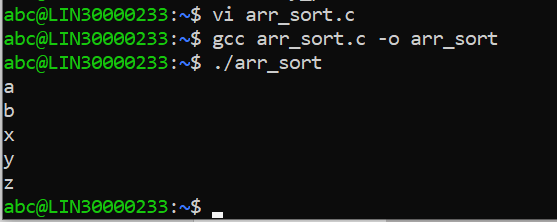
sort(arr, sizeof(arr)/sizeof(arr[0]);

return 0;

}

Ans:





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

int main()

{

char arr[][3] = {

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

return 0;

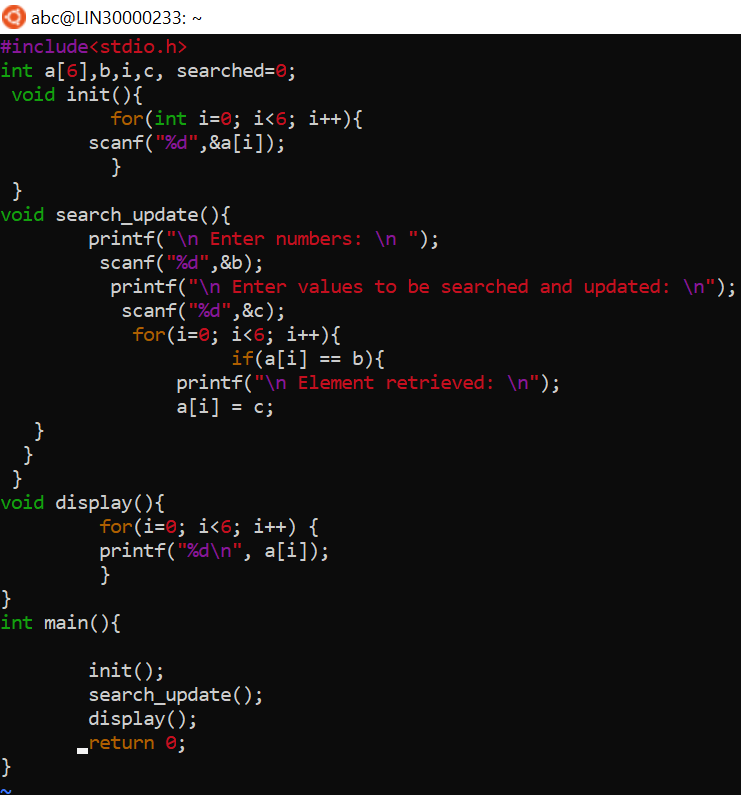
}

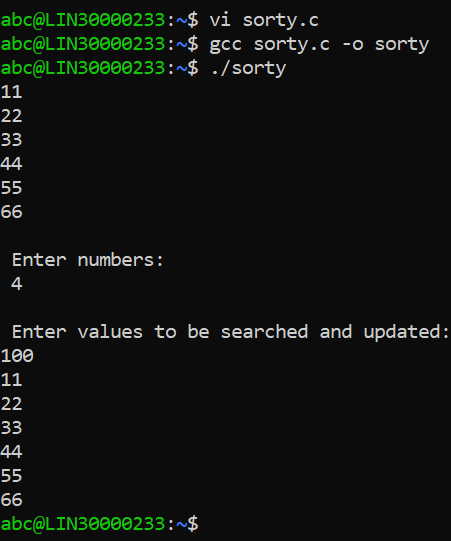
**Allow user to perform the following operations.**

* 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**

Ans:



****