# 1D, 2D, MultiDimensional Array Assignments

**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, we can use pointer in array style in that case it will return the address of that particular element.

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

Ans: Yes, it will return the value of that element.

Eg\*array will return array[0] value,\*(array+10) will return array[10] value

Q3: is ptr++ valid?

ANS: YES

Q4: is array++ valid?

ANS: NO

Q5: what is sizeof(array)?

ANS: It will depend on the array declaration

Q6: what is sizeof(ptr)?

ANS: 8 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}; **// it will assign first 3 values**

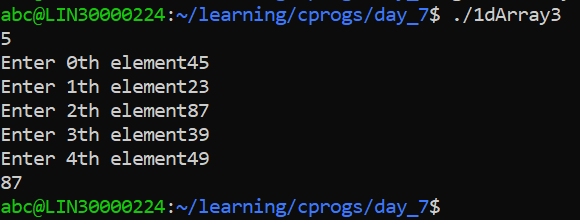
int arr1[MAX]={0}; **// it will assign 0**

static int arr2[MAX];

}

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.





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()

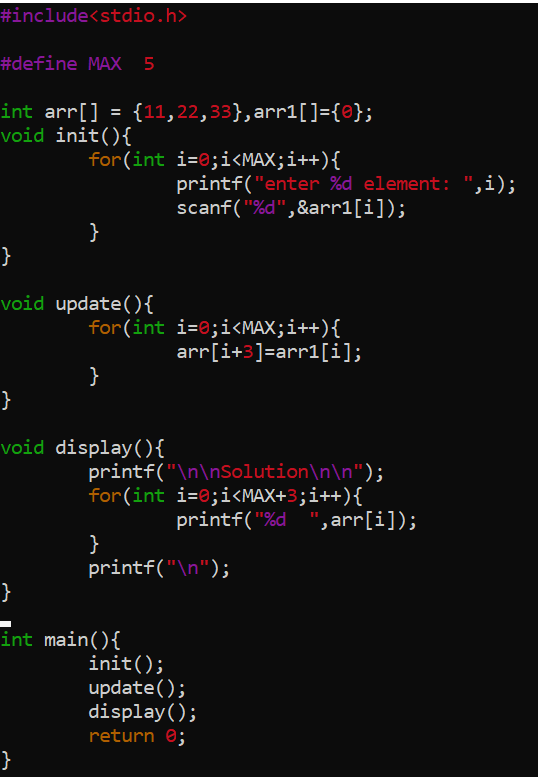
{

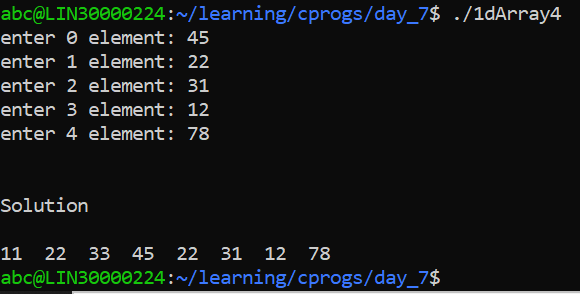
int arr[MAX] = {11,22,33};

}

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





**2D, MultiDimensional Arrays**

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

int main()

{

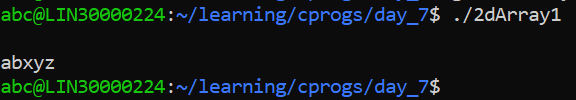
char arr[]= “xaybz”;

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

return 0;

}

****

****

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



