# 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

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

Q3: is ptr++ valid?

Ans: yes it is valid

Q4: is array++ valid?

Ans: no array ++ is not valid (it is a constant pointer)

Q5: what is sizeof(array)?

Ans: it number of elements multiplied by size of the data type

Eg: int arr[10] -> size is 4\*10 = 40bytes

Q6: what is sizeof(ptr)?

Ans: size of pointer will 4/8 bytes based on bittness of the system

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: arr, arr1 are initialized. As size is defined other indexes are stored with zero. Arr2 is declared but not initialized.

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:

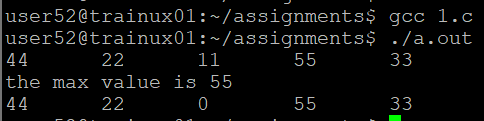
A screen shot of a computer code

Description automatically generated

A black background with blue and red text

Description automatically generated

output:



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.

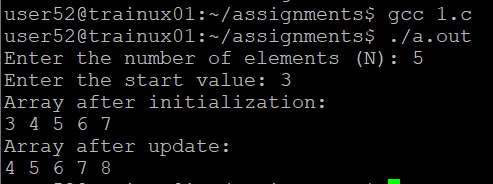
1. init() - Use the inputs to initialize the first N elements of the array with N consequetive values starting with given start value .
2. update() – increment value of every element in the array
3. display() – display the contents of array

A screenshot of a computer program

Description automatically generated

A computer screen shot of a program code

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;

}

Ans:

A screen shot of a computer program

Description automatically generated

A black background with white text

Description automatically generated

1. Refer the code snippet below.

int main ()

{

char arr[][3] = {

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

return 0;

}

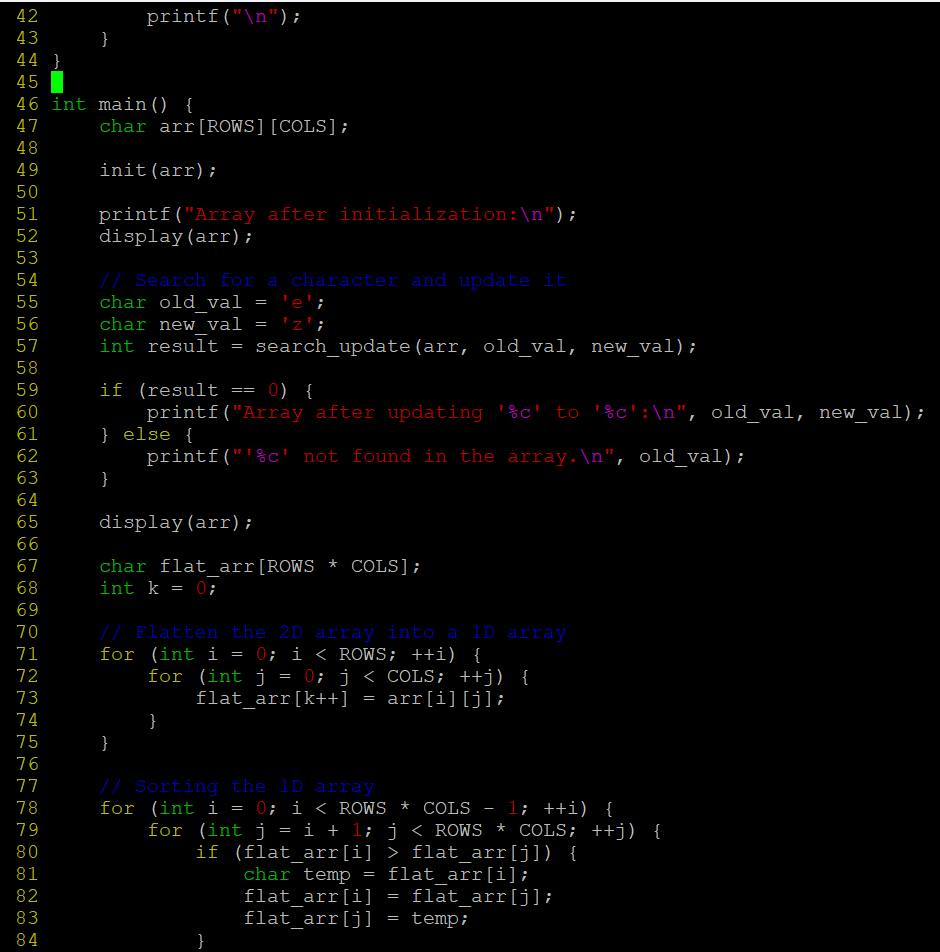
Allow user to perform the following operations.

* 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 computer screen shot of text

Description automatically generated

A screen shot of a computer

Description automatically generated