# 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]

1. Yes. As in array[100] array stores base address and if array is assigned to ptr, we can use ptr in array style.

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

1. Yes.

Q3: is ptr++ valid?

1. Yes. It helps to add number of bytes (based on datatype pointer points) once.

Q4: is array++ valid?

1. Yes.

Q5: what is sizeof(array)?

1. 200 bytes.

Q6: what is sizeof(ptr)?

1. 4 or 8 bytes.
2. 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];

}

1. 1. int arr[MAX] ={11,22,33}; Remaining elements of arr are initialized to 0 by default.

2. int arr1[MAX]= {0}; Remaining elements of arr1 are initialized to 0 by default.

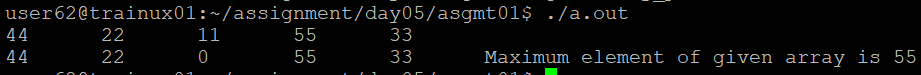
3. static int arr2[MAX]; declares an array with static storage duration, meaning it is initialized once and its elements are initialized to 0 by default.

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.
2. Added and called in main() in array\_pointer.c:

A computer screen shot of 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};

}

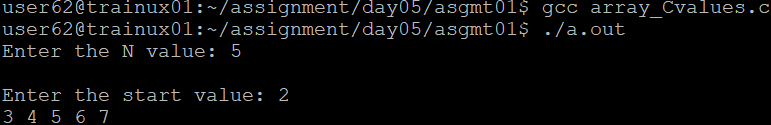
Add the following functions choosing proper input, output and return.

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
4. Code:

A screen shot of a computer program

Description automatically generated

Output:



**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. Code:

****

Output:

A black screen 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

1. Code:

A screen shot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

Output:

A black background with white text

Description automatically generated