**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

}

**1:** Can pointer be used in Array-style syntax?  
**A:** Yes, pointers can be used in array-style syntax, e.g., ptr[10] and ptr[0] are valid.

**2:** Can Array be used in Pointer-style syntax?  
**2:** Yes, arrays can be used in pointer-style syntax, e.g., \*array, \*(array + 0), and \*(array + 10) are valid.

**3:** Is ptr++ valid?  
**3:** Yes, ptr++ is valid as long as ptr is not pointing to read-only memory.

**4:** Is array++ valid?  
**4:** No, array++ is not valid because the name of the array is a constant pointer and cannot be incremented.

**5:** What is sizeof(array)?  
**5:** sizeof(array) is 400 bytes (size of 100 integers, assuming sizeof(int) = 4).

**6:** What is sizeof(ptr)?  
**6:** sizeof(ptr) is the size of a pointer, typically 8 bytes on a 64-bit 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];

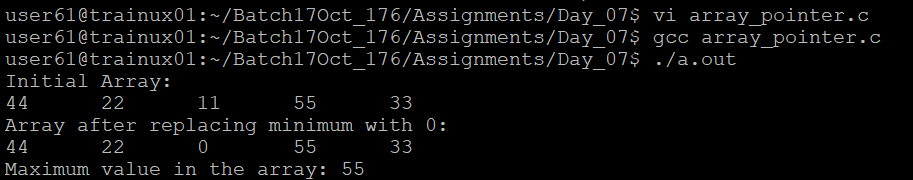
}

* **arr:** Elements other than explicitly initialized ones are set to 0.
* **arr1:** All elements are explicitly initialized to 0.
* **arr2:** Being a static array, all 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.

A screenshot of a computer program

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.

a. init() - Use the inputs to initialize the first N elements of the array with N consequetive values starting with given start value .

b. update() – increment value of every element in the array

c. display() – display the contents of array

A computer screen shot of a program

Description automatically generated

A computer screen with numbers and letters

Description automatically generated

**2D, MultiDimensional Arrays**

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

int main()

{

char arr[]= “xaybz”;

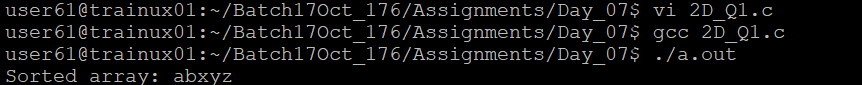
sort(arr, sizeof(arr)/sizeof(arr[0]);

return 0;

}

A computer screen shot of a program code

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.

a. init() - initialize the array and return 0

b. search\_update() – search for a given element in array and if found update it to given value and return 0 else return 1

c. display() – traverse and display array contents

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

A screen shot of a computer program

Description automatically generated

A computer screen with white text

Description automatically generated