**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]

A: Yes, pointers can be used with array-style syntax (ptr[10], ptr[0]).

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

A: **Yes**, arrays can be used with pointer-style syntax (\*array, \*(array + 0), \*(array + 10))

Q3: is ptr++ valid?

A: **Yes**, ptr++ is valid, as it increments the pointer to the next element.

Q4: is array++ valid?

A: **No**, array++ is not valid because arrays are not modifiable lvalues.

Q5: what is sizeof(array)?

A: **400 bytes** (assuming sizeof(int) is 4 bytes).

Q6: what is sizeof(ptr)?

A: **4 bytes** (on a 32-bit system, 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];

}

* First three elements (arr[0], arr[1], arr[2]) are set to 11, 22, and 33 respectively.
* Remaining elements (arr[3] to arr[99]) are automatically set to 0.

arr1[MAX] = {0};

* All elements of arr1 are initialized to 0, as only the first element is explicitly initialized, and the rest are implicitly set to 0.

static int arr2[MAX];

* Since the array is declared as static, all 100 elements of arr2 are automatically initialized to 0 (static variables are automatically initialized to zero if not explicitly 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.

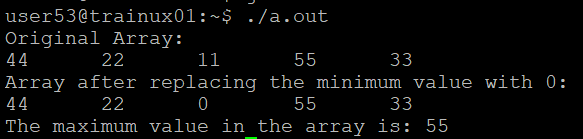
A screen shot of a computer program

Description automatically generated

A black screen with 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()

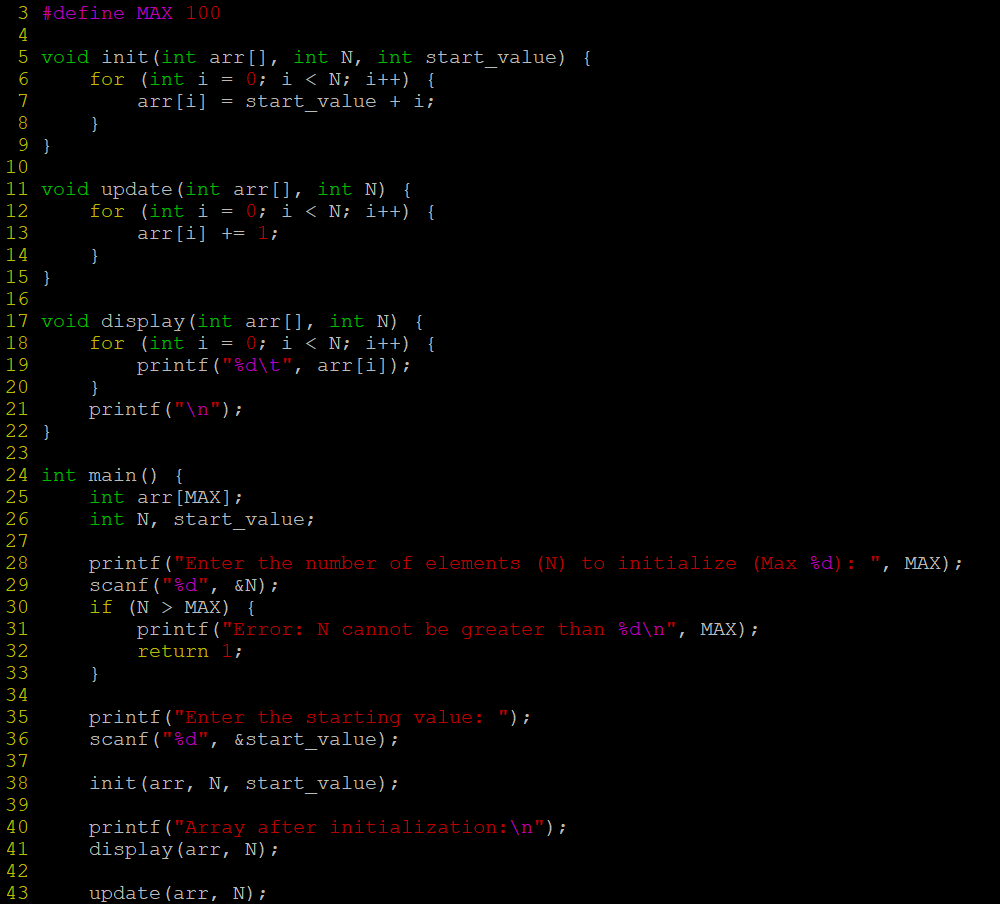
{

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 screen shot of a black background

Description automatically generated

OUTPUT:

A black screen with white text

Description automatically generated

**2D, MultiDimensional Arrays**

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

int main()

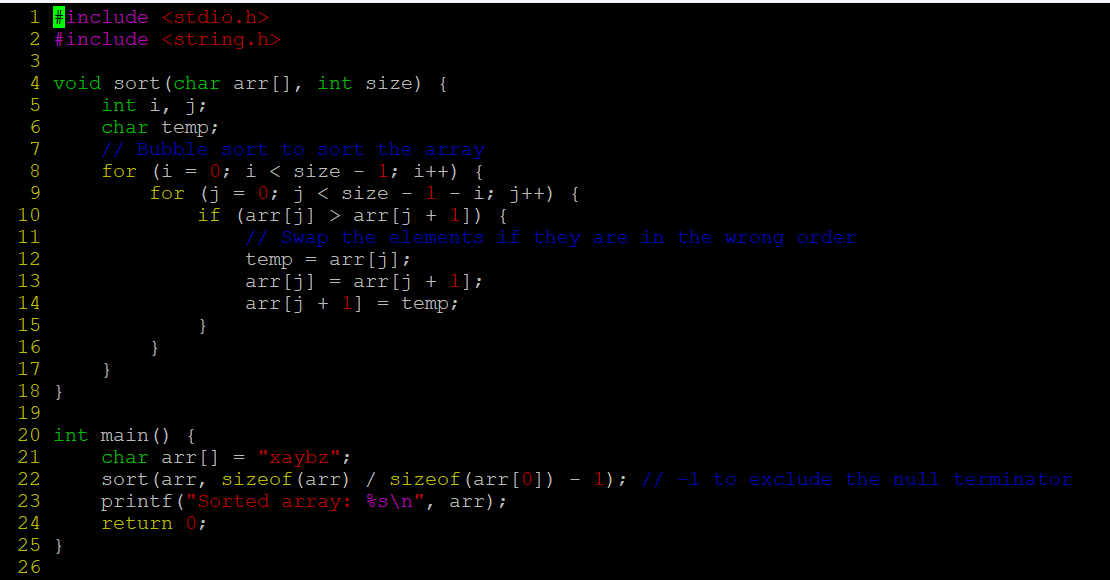
{

char arr[]= “xaybz”;

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

return 0;

}

****

OUTPUT:

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;

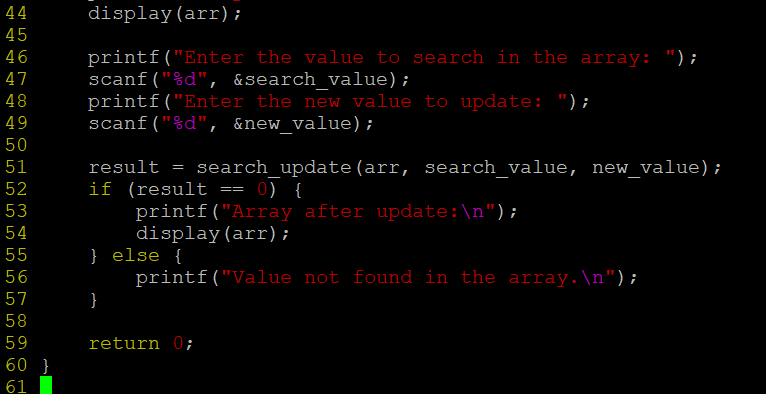
}

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





OUTPUT:

