**1D, 2D, MULTIDIMENSIONAL ARRAY**

**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]

· Yes, a pointer can be used in array-style syntax. For example, if ‘ptr’ points to a valid memory location, we can access elemenrts using ‘ptr[10]’ or ‘ptr[0]’.

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

· Yes, an array can be used in pointer-style syntax. The name of the array can be treated as a pointer to its first element.

Q3: is ptr++ valid?

· Yes, ‘ptr++’ is valid as long as ‘ptr’ is a pointer that has been initialized to point to a valid memory location. This operation increments the pointer to point to the next element of the type it points to.

Q4: is array++ valid?

· No, ‘array++’ is not valid. The name of an array in C is not a modifiable value.

Q5: what is sizeof(array)?

· The ‘sizeof(array)’ will return the total size in bytes of the array.

Q6: what is sizeof(ptr)?

· The ‘sizeof(ptr)’ will return the size of the pointer itself, not the size of the data it points to. Mostly it gives 8bytes or 4 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];

}

Ø Int arr[MAX] ={11,22,33};

· In this array, only the first three elements are explicitly initialized to 11,12, and 33. Remaing elements(97): initialized to 0.

Ø Arr1

· All elements(100): explicity initialized to 0.

Ø Arr2

· All elements(100): initialized to 0

3. 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:

A black screen with white text

Description automatically generated

A screenshot of a computer code

Description automatically generated

4. 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:

A black background with white text

Description automatically generated

A screen shot of a computer program

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:

A black screen with white text

Description automatically generated

A screen shot of a computer code

Description automatically generated

2. 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:



