Assignment 4 Part 1

NSCC | W0400010

PROG2007

Gabriel Levis

2024

Table of Contents

[Part 1: Written Response Questions 2](#_Toc163039359)

[Array Basics 2](#_Toc163039360)

[String Manipulation 3](#_Toc163039361)

[Pointers 5](#_Toc163039362)

[Memory Allocation 6](#_Toc163039363)

[Common Pitfalls 7](#_Toc163039364)

# Part 1: Written Response Questions

## Array Basics

C Array vs. Java and JavaScript Arrays:

In C, arrays are fixed-size collections of elements of the same data type stored in contiguous memory locations. They do not have built-in bound checking.

In Java, arrays are objects that can dynamically change size. They have built-in bounds checking and provide various utility methods for manipulation.

In JavaScript, arrays are dynamic and can hold elements of different data types. They provide flexible methods for manipulation and do not have fixed sizes.

Sources:

<https://stackoverflow.com/questions/3802470/array-string-difference-in-java-vs-c#:~:text=Note%20that%2C%20in%20Java%2C%20arr,until%20a%20complete%20declaration%20appears.&text=Array%20objects%20in%20Java%20must,int%5B10%5D%5B20%5D%3B>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array>

Declaring and Initializing Arrays in C:

To declare an array in C, you specify the data type of the elements and the size of the array.

Arrays in C can be initialized at the time of declaration using curly braces, or initialized later by assigning values to individual elements.

Sources:

<https://www.geeksforgeeks.org/c-arrays/>

<https://www.tutorialspoint.com/cprogramming/c_arrays.htm>

One-Dimensional vs. Multi-Dimensional Arrays:

A one-dimensional array in C stores elements in a linear sequence.

A multi-dimensional array in C stores elements in a tabular form with rows and columns.

Example:

c

Copy code

// One-Dimensional Array

int arr1D[5] = {1, 2, 3, 4, 5};

// Multi-Dimensional Array

int arr2D[3][3] = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

Sources:

<https://www.programiz.com/c-programming/c-arrays>

<https://www.tutorialspoint.com/cprogramming/c_multi_dimensional_arrays.htm#:~:text=Multi−dimensional%20arrays%20can%20be,forms%20a%20two−dimensional%20array>

## String Manipulation

Representation of Strings in C:

In C, strings are represented as arrays of characters terminated by a null character '\0'.

Strings in C are not a separate data type but are arrays of characters.

Sources:

<https://www.geeksforgeeks.org/strings-in-c/>

<https://www.tutorialspoint.com/cprogramming/c_strings.htm>

strcpy Function:

The strcpy function in C is used to copy one string to another.

It takes two arguments: destination string and source string.

Example:

c

Copy code

char destination[20];

char source[] = "Hello";

strcpy(destination, source);

Sources:

<https://www.programiz.com/c-programming/library-function/string.h/strcpy#:~:text=The%20function%20prototype%20of%20strcpy,also%20returns%20the%20copied%20string>.

<https://www.tutorialspoint.com/c_standard_library/c_function_strcpy.htm>

Another Function from String.h:

Another function from the string.h library is strlen, which is used to find the length of a string.

Example:

c

Copy code

char str[] = "Hello";

int length = strlen(str); // length will be 5

Sources:

<https://www.geeksforgeeks.org/php-strlen-function/>

<https://www.tutorialspoint.com/c_standard_library/c_function_strlen.htm>

## Pointers

Definition of a Pointer in C:

A pointer in C is a variable that stores the memory address of another variable.

Pointers are used to indirectly access the memory location and manipulate data stored there.

Sources:

<https://www.geeksforgeeks.org/c-pointers/>

<https://www.tutorialspoint.com/cprogramming/c_pointers.htm#:~:text=Pointer%20is%20a%20variable%20that,even%20a%20pointer%20type%20itself>.

Declaring and Initializing a Pointer Variable:

To declare a pointer variable in C, you specify the data type followed by an asterisk (\*) before the variable name.

You can initialize a pointer variable by assigning the address of another variable to it.

Example:

c

Copy code

int \*ptr; // Pointer declaration

int num = 10;

ptr = &num; // Pointer initialization

Sources:

<https://www.programiz.com/c-programming/c-pointers>

<https://www.tutorialspoint.com/cprogramming/c_pointers.htm>

Pointer Arithmetic:

Pointer arithmetic in C allows performing arithmetic operations on pointer variables.

When you perform arithmetic on pointers, the size of the data type they point to determines the increment or decrement in their address.

Sources:

<https://www.geeksforgeeks.org/pointer-arithmetics-in-c-with-examples/>

<https://www.tutorialspoint.com/cprogramming/c_pointer_arithmetic.htm>

## Memory Allocation

Dynamic Memory Allocation:

Dynamic memory allocation in C allows the program to allocate memory dynamically during runtime.

It is achieved using functions like malloc, calloc, and realloc.

Sources:

<https://www.geeksforgeeks.org/what-is-dynamic-memory-allocation/>

<https://www.tutorialspoint.com/what-is-dynamic-memory-allocation-in-c>

Comparison of malloc, calloc, and realloc:

malloc allocates a block of memory of specified size.

calloc allocates a block of memory and initializes all its bits to zero.

realloc changes the size of the previously allocated memory block.

Sources:

<https://www.programiz.com/c-programming/c-dynamic-memory-allocation>

<https://www.tutorialspoint.com/what-is-dynamic-memory-allocation-in-c>

## Common Pitfalls

Common Mistakes:

Common mistakes include dereferencing uninitialized pointers, accessing arrays out of bounds, and forgetting to null-terminate strings.

Another mistake is not deallocating dynamically allocated memory, leading to memory leaks.

Sources:

<https://www.geeksforgeeks.org/errors-in-cc/>