C Programming Practice Questions: Arrays, Strings, and Pointers

Arrays

Introduction to Arrays: Q1: Write a C program to store and print the first 10 natural numbers using an array.

Introduction to Arrays: Q2: Explain the difference between declaring int arr[10]; and int *arr; in terms of memory allocation.

Single-Dimensional Arrays: Q1: Write a program to find the maximum and minimum elements in an array of n integers.

Single-Dimensional Arrays: Q2: Write a program to reverse the elements of a 1D array in place.

Advanced: Write a program to rotate an array by k positions without using an extra array.

Advanced: Implement a binary search in a sorted array using recursion.

Multidimensional Arrays: Q1: Write a program to add two 3×3 matrices.

Multidimensional Arrays: Q2: Write a program to search for a specific element in a 2D array.

Advanced: Implement matrix multiplication for two matrices of compatible dimensions.

Advanced: Write a program to find the transpose of a matrix in place.

Strings in C

Strings in C: Q1: Write a program to count the number of vowels in a given string without using library functions.

Strings in C: Q2: Explain the difference between a character array (char str[]) and a pointer to char (char *str).

String Handling Functions: Q1: Write a program to check whether a given string is a palindrome using strlen() and strcmp().

String Handling Functions: Q2: Write a program to concatenate two strings using strcat().

Advanced: Implement your own strlen(), strcpy(), and strcmp() functions without using the string.h library.

Array of Strings: Q1: Write a program to sort an array of 5 strings in alphabetical order.

Array of Strings: Q2: Write a program to search for a name in an array of names using strcmp().

Advanced: Write a program to find the longest string in an array of strings.

Pointers

Introduction to Pointers: Q1: Write a program to demonstrate how a pointer can be used to access and modify the value of a variable.

Introduction to Pointers: Q2: Explain what happens when you assign one pointer to another in C.

Pointer Arithmetic: Q1: Write a program that uses pointer arithmetic to traverse and print an integer array.

Pointer Arithmetic: Q2: Write a program to copy elements from one array to another using only pointers.

Pointers and Arrays: Q1: Write a program to find the sum of elements of an array using pointers.

Pointers and Arrays: Q2: Write a program to reverse a string using pointers.

Pointers and Strings: Q1: Write a program to count the number of words in a string using pointers.

Pointers and Strings: Q2: Write a program to replace all spaces in a string with '-' using pointers.

Pointers to Pointers: Q1: Write a program to demonstrate accessing a 2D array using a pointer to a pointer.

Pointers to Pointers: Q2: Explain with code how char **argv works in the main() function for command-line arguments.

Advanced: Write a program to dynamically create a 2D array using pointers to pointers and free its memory.

Advanced: Implement a function that swaps two strings using pointers.

Dynamic Memory Allocation

malloc, calloc, realloc, free: Q1: Write a program to dynamically allocate memory for n integers using malloc, take input, and print them.

malloc, calloc, realloc, free: Q2: Write a program to read n integers using calloc, then use realloc to increase the size by 5 and print the updated array.

Advanced: Write a program to dynamically allocate memory for a matrix and perform matrix addition.

Advanced: Implement a program to dynamically store multiple strings using malloc and free all allocated memory at the end.