

Week 4 Practice Tasks

Website: www.zypheron.tech

Contact: info@zypheron.tech

Practice Problem Set: C Programming (Functions, Arrays, Operators, Loops - Mixed)

1. **Even Number Summation:** Develop a C function that accepts an integer array and its size. The function should compute and return the sum of all even numbers within the array, utilizing a loop and the modulus operator for parity checks.
2. **Maximum Value Determination:** Implement a function that takes an array and its size as input. This function should identify and return the maximum value present in the array, employing a loop and comparison operators.
3. **Positive Number Count:** Create a C program that prompts the user to input 10 integers into an array. Subsequently, use a function to count and return the number of positive integers within the array, leveraging a loop and relational operators.
4. **In-Place Array Reversal:** Design a function to reverse the elements of an integer array. The function should modify the array in place, achieving the reversal through iterative processing with a loop.
5. **Student Average Marks Calculation:** Write a C program that reads the marks of 'N' students into an array, where 'N' is user-defined. Utilize a function to calculate and display the average marks, employing appropriate arithmetic operators for computation.
6. **Element-Wise Array Summation:** Construct a function that accepts two integer arrays of equal size and their common size. The function should return a new array where each element is the sum of the corresponding elements from the input arrays, using a loop and arithmetic operators.
7. **Integer Presence Check:** Develop a function that verifies the presence of a user-provided integer within a given array. The function should return a boolean indicator (true/false) based on the search result, using a loop and logical operators.
8. **Second Largest Element Identification:** Given an array of integers, write a function to identify and return the second largest element. This task must be accomplished using only loops and relational operators, without recourse to built-in sorting or statistical functions.
9. **Conditional Odd Number Printing:** Implement a program using functions and loops to print all odd numbers from 1 to 100. However, numbers that are divisible by both 3 and 5 must be skipped. Employ logical and comparison operators for filtering.
10. **Array Right Shift:** Create a C program with a function to perform a right circular shift on an integer array by one position (the last element moves to the front). A loop should be used to execute this shift operation.