CPP-A10: Pointer_Part-2

Task 1: Reverse a String Using Pointers

Problem Statement:

Write a function to reverse a string using pointers. The program should accept a string and print the reversed version.

Input:

A single string (e.g., `"HelloWorld"`).

Output:

Reversed string (e.g., `"dlroWolleH"`).

Task 2: Count Vowels and Consonants Using Pointers

Problem Statement:

Create a function that counts the number of vowels and consonants in a given string using pointers.

Input:

A string (e.g., `"Masai School"`).

Output:

```
Vowels: 5
Consonants: 7
```

Task 3: Check for Palindrome Using Pointers

Problem Statement:

Write a function to check whether a given string is a palindrome using pointers.

Input:

A string (e.g., `"radar"`).

Output:

```
The string is a palindrome.
```

OR

```
The string is not a palindrome.
```

Task 4: Find the Length of a String Using Pointers

Problem Statement:

Write a function to calculate the length of a string using pointers without using `strlen()`.

Input:

A string (e.g., `"CodingIsFun"`).

Output:

```
Length of string: 11
```

Task 5: Copy One String to Another Using Pointers

Problem Statement:

Create a function to copy one string to another character array using pointers.

Input:

A string (e.g., `"Masai"`).

Output:

```
Original String: Masai
Copied String: Masai
```

Task 6: Compare Two Strings Using Pointers

Problem Statement:

Write a function to compare two strings character by character using pointers without built-in functions.

Input:

Two strings (e.g., `"apple"` and `"applepie"`).

Output:

```
First string is smaller.
```

Task 7: Convert Uppercase Letters to Lowercase (and Vice Versa) Using Pointers

Problem Statement:

Write a function to convert uppercase characters to lowercase and vice versa using pointers.

Input:

A string (e.g., `"HelloWorld"`).

Output:

```
Converted String: hELLOwORLD
```

Task 8: Remove Duplicate Characters from String Using Pointers

Problem Statement:

Create a function to remove duplicate characters from a given string using pointers.

Input:

A string (e.g., `"programming"`).

Output:

```
String without duplicates: progamin
```

Task 9: Count Words in a Sentence Using Pointers

Problem Statement:

Write a function to count the number of words in a sentence using pointers.

Input:

A sentence (e.g., `"Hello world from Masai"`).

Output:

```
Word count: 4
```

Task 10: Find Substring Occurrences Using Pointers

Problem Statement:

Write a function to find and count occurrences of a given substring using pointers.

Input:

```
Main string: HelloHelloHello
Substring: Hello
```

Output:

```
Occurrences of substring: 3
```

Additional Advanced Problems

Task 11: Merge Two Sorted Arrays

Problem Statement:

Write a function to merge two sorted arrays into a single sorted array using pointers.

Input:

```
Array 1: [1, 3, 5]
Array 2: [2, 4, 6]
```

Output:

```
Merged Array: [1, 2, 3, 4, 5, 6]
```

Task 12: Find the Maximum and Minimum in an Array Using Pointers

Problem Statement:

Create functions to find the maximum and minimum values in an array using pointers.

Input:

An array (e.g., `[10, 3, 7, 2, 15]`).

Output:

```
Maximum: 15
Minimum: 2
```

Task 13: Reverse an Array Using Pointers

Problem Statement:

Write a function to reverse an array in place using pointers.

Input:

An array (e.g., `[1, 2, 3, 4, 5]`).

Output:

```
Reversed Array: [5, 4, 3, 2, 1]
```

Task 14: Count Frequencies of Characters in a String Using Pointers

Problem Statement:

Create a function to count the frequency of each character in a string using pointers.

Input:

A string (e.g., `"apple"`).

Output:

```
a: 1
p: 2
```

```
l: 1
e: 1
```

Task 15: Find Common Elements Between Two Arrays Using Pointers

Problem Statement:

Write a function to find and print common elements between two arrays using pointers.

Input:

```
Array 1: [1, 2, 3, 4, 5]
Array 2: [3, 4, 5, 6, 7]
```

Output:

```
Common Elements: 3, 4, 5
```

Task 16: Find Unique Elements in an Array Using Pointers

Problem Statement:

Write a function to find elements that appear only once in an array using pointers.

Input:

An array (e.g., `[1, 2, 2, 3, 4, 4, 5]`).

Output:

```
Unique Elements: 1, 3, 5
```

Task 17: Rotate an Array by K Positions Using Pointers

Problem Statement:

Write a function to rotate an array by `k` positions using pointers.

Input:

```
Array: [1, 2, 3, 4, 5]
K: 2
```

Output:

```
Rotated Array: [4, 5, 1, 2, 3]
```

Task 18: Find the Longest Word in a Sentence Using Pointers

Problem Statement:

Write a function to find the longest word in a sentence using pointers.

Input:

A sentence (e.g., `"Masai School teaches coding"`).

Output:

```
Longest Word: teaches
```

Task 19: Concatenate Two Strings Using Pointers

Problem Statement:

Write a function to concatenate two strings without using built-in functions and using pointers.

Input:

```
String 1: "Hello"
String 2: "World"
```

Output:

```
Concatenated String: HelloWorld
```

Task 20: Check Anagram Strings Using Pointers

Problem Statement:

Write a function to check whether two strings are anagrams of each other using pointers.

Input:

```
String 1: "listen"
String 2: "silent"
```

Output:

```
The strings are anagrams.
```

Happy Coding!