Pointer Based C Programming Practice

Deadline: 13 November, 4:30 PM

Instructions

- Complete each program as per the questions provided.
- All programming should be done in a Linux environment.
- Ensure to test each program for correctness and capture the expected output.
- Save all programs for future use and reference.

1. Pointer Initialization

Question: Write a C program that initializes an integer variable with a value, then uses a pointer to modify that value. Print both the original and modified values.
Expected Output:

```
Original value of a: 10
Modified value of a: 20
```

2. Array Traversal with Pointers

Question: Create a C program that declares an array of integers and uses a pointer to traverse and print each element of the array.
Expected Output:

```
Element at index 0: 10
Element at index 1: 20
Element at index 2: 30
Element at index 3: 40
Element at index 4: 50
```

3. Swapping Values Using Pointers

Question: Write a C function that swaps two integers using pointers. Demonstrate the function by swapping two integer variables and displaying their values before and after the swap.
Expected Output:

```
Before swap: a = 5, b = 10
After swap: a = 10, b = 5
```

4. Dynamic Memory Allocation

Question: Write a C program that dynamically allocates an array of integers, fills it with values, and then frees the allocated memory. Print the values of the array.
Expected Output:

```
10 20 30 40 50
```

5. Pointer to Pointer

Question: Write a C program that demonstrates the concept of a pointer to a pointer. Use it to modify the value of an integer variable.
Expected Output:

```
Original value of a: 20
Modified value of a: 30
```

6. String Manipulation Using Pointers

Question: Create a C program that takes a string input from the user and uses pointers to reverse the string. Print the reversed string.
Expected Output (example with user input):

```
Enter a string: Hello
Reversed string: olleH
```

7. Finding the Length of a String

Question: Write a function that takes a string as input (using a pointer) and returns its length without using the `strlen` function.
Expected Output:

```
Length of the string: 13
```

8. Passing Arrays to Functions

Question: Write a C program that defines a function to find the average of an array of integers. Pass the array to the function using pointers and print the average in `main`.
Expected Output:

```
Average of the array: 3.00
```

9. Copying Strings Using Pointers

Question: Write a function that copies one string to another using pointers. In `main`, demonstrate this function by copying a predefined string and printing the copied string.
Expected Output:

```
Copied String: Goodbye
```

10. Reversing an Array

Question: Write a C program that reverses the contents of an integer array using pointers. The program should take input for the array size and elements, reverse the array, and print the reversed array.
Expected Output (example with user input):

```
Enter the size of the array: 5
Enter 5 elements: 1 2 3 4 5
Reversed array: 5 4 3 2 1
```

11. Matrix Addition Using Pointers

Question: Write a C program that performs matrix addition. The program should take two 2D arrays (matrices) as input, use pointers to add them, and store the result in a third matrix. Print the resulting matrix.
Expected Output (example with user input):

```
Enter the size of the matrices (rows and columns): 2 2
Enter elements of first matrix:
1 2
3 4
Enter elements of second matrix:
5 6
7 8
Resulting Matrix:
```

```
6 8
10 12
```

12. Finding the Maximum Element in an Array Using Pointers

Question: Write a C program that finds the maximum element in an array of integers using pointers. The program should take input for the array size and elements, and print the maximum value found.
Expected Output (example with user input):

```
Enter the size of the array: 5
Enter 5 elements: 10 20 5 30 15
Maximum element: 30
```

13. Sorting an Array Using Pointers

Question: Implement a C program that sorts an array of integers using the bubble sort algorithm. Use pointers to manipulate the elements of the array during the sorting process. Print the sorted array.
Expected Output (example with user input):

```
Enter the size of the array: 5
Enter 5 elements: 64 34 25 12 22
Sorted array: 12 22 25 34 64
```

14. Reversing a String Using Recursion and Pointers

Question: Write a recursive function that reverses a string using pointers. In `main`, call this function and print the reversed string.
Expected Output (example with user input):

```
Enter a string: Hello
Reversed string: olleH
```

15. Counting Vowels in a String Using Pointers

Question: Create a C program that counts the number of vowels in a given string using pointers. The program should take a string input from the user and display the number of vowels found.
Expected Output (example with user input):

```
Enter a string: Hello World
Number of vowels: 3
```