SPOT EXERCISE FOR EXPERIMENT NUMBER 11 AND 12

1. Arrays and Pointers - Find the Second Largest Distinct Element Without Sorting.

The academic section wants to identify the student who has obtained the second-highest distinct mark in a class. The marks are stored in an integer array. Multiple students may have obtained the same mark.

Input:

85 92 76 92 68 85 80

Expected Output:

Highest Mark: 92

Second Highest Distinct Mark: 85

Develop a C program to identify the second-largest distinct element in an array using pointers with the following Constraints

1. Do not sort the array.
2. Do not create another array.
3. Do not use conventional array indexing for accessing elements.
4. Use pointer arithmetic to traverse the array.
5. Handle duplicate maximum values correctly.
6. Display an appropriate message if a second-largest distinct element does not exist.

2. Strings with Pointers - Reverse the Words in a Sentence Without Reversing Individual Words

The CEG digital notice board receives announcements as text. For certain display requirements, the administrator wants to reverse the order of the words in an announcement while retaining the original spelling of every word.

Input:

WELCOME TO COLLEGE OF ENGINEERING GUINDY

Expected Output:

GUINDY ENGINEERING OF COLLEGE TO WELCOME

Develop a C program to reverse the order of words in a given sentence using character pointers with the following constraints

1. Do not use predefined string handling functions such as strlen(), strrev(), strcpy() or strcat().
2. Do not use another character array to store the reversed sentence.
3. Use character pointers for traversing and manipulating the original string.
4. Individual words must retain their original character order.
5. The solution must modify the original string.

3. Dynamic Memory Allocation - Shrink Dynamic Memory Based on Actual Data Requirements

A research laboratory initially allocates memory to store N experimental measurements. After collecting the data, the research team discovers that only measurements satisfying a specified acceptance criterion need to be retained for further analysis. The application should remove the unwanted measurements and reduce the allocated memory according to the actual number of retained measurements.

Original Measurements:
12 45 18 67 25 80 32
Acceptance Criterion:
Measurements >= 30
Expected Retained Measurements:
45 67 80 32
Final Number of Measurements:
4

Develop a C program that dynamically allocates memory for N measurements, retains only those satisfying the given criterion, and reduces the allocated memory accordingly with the following constraints

1. Do not use statically declared arrays.
2. Do not allocate a second dynamic array.
3. Perform filtering within the originally allocated memory block.
4. Use realloc() to reduce the allocated memory to the required size.
5. Use pointers to access and modify the measurements.
6. Release the allocated memory using free().