|  |  |
| --- | --- |
| **TASK 1: 1D Dynamic Arrays** |  |

1. Write a function **int\* InputArray(int& size)** that asks user to enter size of required array, allocates the memory on heap, takes input in array and returns its pointer.
2. Write a function **void OutputArray(const int\* myArray, const int& size)** that takes a pointer to an integer array and prints its data. Write main function to test above functionality.

Why is the parameter size passed as const? Write your answer in comments

1. Write a function **int\* CompressArray(int\* oiginalArr, int& size)** that takes a sorted array and removes duplicate elements from this array.

**Sample Run:**

**//Input:**

Enter Size of array: 10

Enter 10 elements: 1 2 2 2 3 3 3 3 3 7

**//Output**

Array after Compression: 1 2 3 7

Your function will compress the original array, allocate new array of compressed size (compressed size is 4 in above example) on heap, copy updated array in new array and return the new array. Take input from user by calling int\* InputArray(int& size) (function you implemented in Task 1 part a). Call CompressArray, call OutputArray( function you implemented in Task 1 part b) to display the final output.

1. Implement a function **int\* Union(int\* setA, int& size1, int\* setB, int& size2, int& size3)** that finds union(common elements) of two sets (stored using arrays).

**Sample Run:**

**//Input:**

Enter Size of Array: 6

Enter 6 elements: 1 2 3 4 5 6

Enter Size of Array: 4

Enter 3 elements: 1 3 5 7

**//Output**

A = {1,2,3,4,5,6}

B = {1,3,5,7}

A union B = {1,2,3,4,5,6,7}

Array3 should not have any duplicate elements. Do not sort array3; instead, find union in such a way that it is sorted by default.

You have to:

* Allocate the three arrays dynamically after inputting the size of array1 and array2 from the user. Statically allocated arrays are NOT allowed
* Initially you can allocate elements = (size of array1 + size of array2) to array3. For example, you would allocate 6+4 to array3 for the above example. After finding the distinct elements from both arrays, the previously allocated size of array3 may be more than what you actually need. (In the above example you require size 7 whereas you have allocated 10).
* Make sure to clear the memory at the end and there should be no memory leak.

|  |  |
| --- | --- |
| TASK 2: Static Character Array |  |

Write a function ***“Counter”*** that receives a character array and calculates Total characters, uppercases, lowercases, vowels, consonants and spaces then return them all. (In this question, all characters except spaces are counted as letters).

**Sample Run:**

|  |
| --- |
| **Character Array:**  Hello, I am a stu dent of FAST  University.  **Values returned:**  No. of letters are: 34  No. of spaces are: 7  No. of uppercase letters are: 7  No. of lowercase letters: 25  No. of Vowels: 13  No. of Consonants: 19 |

|  |  |
| --- | --- |
| TASK 3: Dynamic Character Array |  |

Write C++ program which creates a dynamic character array.

* Ask the user to enter a string in that array.
* Ask user to enter a substring and search either that substring is present in character array or not.
* If the substring is not present ask user to enter the input again unless user enters the substring present in an array.

**For Example:**

**Input:**

Please enter string: **My name is abc**

**Output:**

Please enter substring: xyz

Sorry entered substring is not present in array.

Please enter substring: abc

The entered substring is present in array.