Analyzing Time & Space Complexity - Exercise

For all the problems given below, implement the code and write its time and space complexity. Try to develop the most optimal code in terms of time and space complexity.

Problem 1:

Given an array A of ‘n’ integers, having values in the range 1 to n + 1, find the number missing in the array, considering that there are no duplicate elements in the array.

Input: Array A of size n

Output: missing number in the range 1 to n + 1

Sample: input : A = {6,1,2,8,3,4,7,10,5}, output: 9

Follow up: How will the program have to be modified if the range starts from k?

Problem 2:

Given an array A of ‘n + 2’ integers, having values in the range 1 to n, find the two numbers repeating in the array.

Input: Array A of size n + 2

Output: two numbers occurring twice in A

Sample: input: A = {4, 2, 4, 5, 2, 3, 1}, output: 4 2

Follow up: How will the program have to be modified if the range starts from k?

Problem 3:

Given an array A of ‘n’ positive integers, find the indices of elements which are greater than or equal to all elements on their right. If there is no such index, return -1.

Input: Array A of size n

Output: indices of elements which are greater than or equal to all elements on the their right

Sample: input: {16,17,4,3,5,2}, output: 2, 5, 6

Follow up: Will the same program work, if the input array contains negative numbers as well?

Problem 4:

Given an array A of ‘n’ positive integers, find the index of the element whose left side and right side elements sum to the same value. If there are multiple such indices, return the first of its kind. If there is no such index, return -1.

Input: Array A of size n

Output: first index such that sum of elements on the left side is equal to the sum of elements on the right side

Sample: input: {1,3,5,2,2}, output: 2

Follow up: Will the same program work, if the input array contains negative numbers as well?

Problem 5:

A processor is designed in a way that it has only 3 operations possible: subtraction by 1, division by 2, division by 3, and negation. Write a function (dropTo1) to find the minimum number of operations performed by such a processor to convert a number ‘n’ to 1. [Note: If a number cannot be converted to 1, the function should return -1]

Sample:

Input: 1; Output: 0
Input: 2; Output: 1
Input: -2; Output: 2
Input: 4; Output: 2
Input: 5; Output: 3
Input: -6; Output: 3
Input: 7; Output: 3
Input: 8; Output: 3
Input: 9; Output: 2
Input: -10; Output: 4

Problem 6:

Given two character arrays representing two strings / sentences, write a function to check if one string can be obtained by rearranging the characters in another string. Note: In the process, special characters including space, and case of the characters can be ignored.

Function Declaration: int sameOnRearranging(char str1[], int len1, char str2[], int len2);
Here, len1 and len2 are respectively the lengths of strings str1 and str2.

Sample:

Input: “asd”, 3, “DSA”, 3 Output: 1 ➔ indicates true

Input: “tight”, 5, “right”, 5 Output: 0 ➔ indicates false
Input: “Tom Marvolo Riddle”, 18, “I am Lord Voldemort”, 19 Output: 1
Input: “Apple”, 5, “leap”, 4 Output: 0
Input: “little”, 5, “little”, 5 Output: 1

<u>Problem 7:</u>

Create a structure with pointers to represent a library of books. The Book structure consists of title, author, year of publishing. Create a structure pointer “library”’ using dynamic memory allocation to aid in performing various operations.

Implement functions to perform the following operations:

1. Add new book details
2. Search for books by title
3. Update book information by title
4. Delete books by title
5. Display total number of books in the library
6. Display books published in a specific year