Logical questions for practice

String:

1. Print the non repeating character in a string.
2. In a given string s find the length of the longest substring without repeating characters.
3. Given a string of characters, find the length of the longest proper prefix which is also a proper suffix.
4. Implement a program to validate an IP Address (The IP address is given in string format).
5. Create an array of strings. Write a program to remove all the empty strings from that list.
6. Create an array of strings. Write a program to remove the strings with the lowest length.
7. Check if the given 2 strings are anagrams of each other.

Arrays:

1. Write a program to find the maximum and minimum value of an array
2. Write a program to check if the array is sorted in ascending order or descending order.
3. Write a program to remove a specific element from an array.
4. Find the element that has the maximum frequency in the array
5. Given an array or integers, retain only those integers which appear only once and remove all the remaining elements from the array.
6. Given 2 arrays, find the elements of the second which are missing from the first.
7. Find the second non-repeating element in the array
8. Segregate all the negative and positive integers in the array without changing their relative position.

Complex problems:

1. Write a program to print all of the permutations in a string.
2. Find the longest palindrome in string.
3. Given a string 's'. The task is to find the smallest window length that contains all the characters of the given string at least one time. (For eg. A = aabcbcdbca, then the result would be 4 as of the smallest window will be dbca.)

4. You are given any 2 strings, find the length of the longest common substring in them.
5. Given an integer array, sort its element by their frequency and index. i.e., if two elements have different frequencies, then the one which has more frequency should come first; otherwise, the one which has less index should come first.

For example,
Input : [3, 3, 1, 1, 1, 8, 3, 6, 8, 7, 8]

Output: [3, 3, 3, 1, 1, 1, 8, 8, 8, 6, 7]

6. Given two integer arrays, reorder elements of the first array by the order of elements defined by the second array.

The elements that are not present in the second array but present in the first array should be appended at the end sorted. The second array can contain some extra elements which are not part of the first array.

For example,
Input:

first = [5, 8, 9, 3, 5, 7, 1, 3, 4, 9, 3, 5, 1, 8, 4]
second = [3, 5, 7, 2]

Output: [3, 3, 3, 5, 5, 5, 7, 1, 1, 4, 4, 8, 8, 9, 9]

7. Given an integer array with many duplicated elements, write an algorithm to efficiently sort it in linear time, where the order of equal elements doesn’t matter.

For example,
Input: { 4, 2, 40, 10, 10, 1, 4, 2, 1, 10, 40 }

Output: { 1, 1, 2, 2, 4, 4, 10, 10, 10, 40, 40 }

8. Given an array of positive and negative integers, segregate them without changing the relative order of elements. The output should contain all positive numbers follow negative numbers while maintaining the same relative ordering.

For example,
Input: [9, -3, 5, -2, -8, -6, 1, 3]
Output: [-3, -2, -8, -6, 9, 5, 1, 3]

9. You are given n pairs of strings. Two pairs (a,b) and (c,d) are identical if a==b and c==d . After taking each pair as input, you need to print number of unique pairs you currently have.

sample input
5
john tom
john mary
john tom
mary anna
mary anna

sample output
1
2
2
3
3

Data structures

Implement a stack of integers. Use classes to implement this.
Implement a priority of tasks. Use classes to implement this.
Given n elements of a stack ‘st’ where the first value is the bottom-most value of the stack and the last one is the element at top of the stack, delete the middle element of the stack without using any additional data structure.