Array

Basic Questions:

1) **Create an Array of size 10 of integers. Take input from the user for these 10 elements and print the entire array after that.**

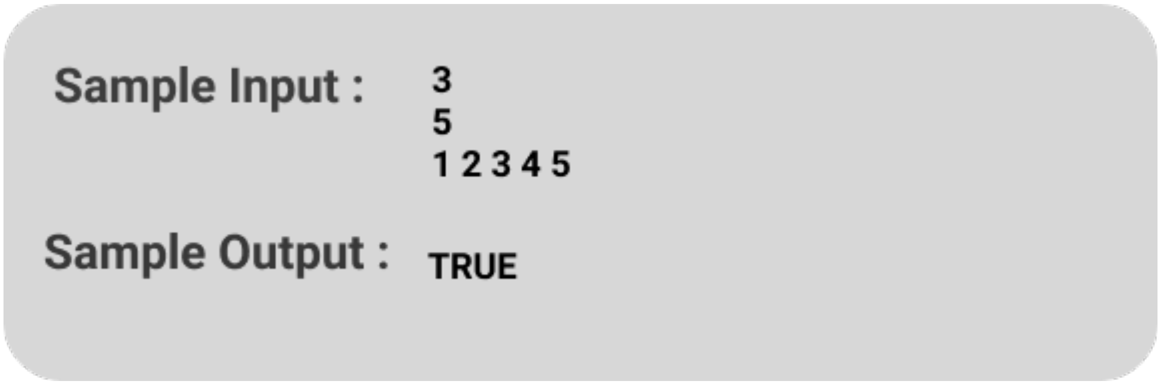
2) **Check whether n is present in an array of size m or not.**

Input - n,m (Input number, size of array)

- Take input n elements for the array

Output -> true/false

[Hint : - Create a dynamic array]



Sample Input :

```
3
5
1 2 3 4 5
```

Sample Output :

```
TRUE
```

3) **Find the minimum and maximum element in an array.**

[Solution: https://www.geeksforgeeks.org/program-find-minimum-maximum-element-array/]

```
Sample Input :     5
                   1 2 3 4 5

Sample Output :    MAX : 5
                   MIN : 1
```

Here in sample input : 5 is the size of array

4) **Write a program to reverse the array.**

[Hint: use indexes]

Here , 5 is the size of array and then elements of arrays are input

```
Sample Input :     5
                   1 2 3 4 5

Sample Output :    5 4 3 2 1
```

5) **Write a program to sort the given array.**

[Hint: use any sorting algorithm i.e. https://www.geeksforgeeks.org/sorting-algorithms/]

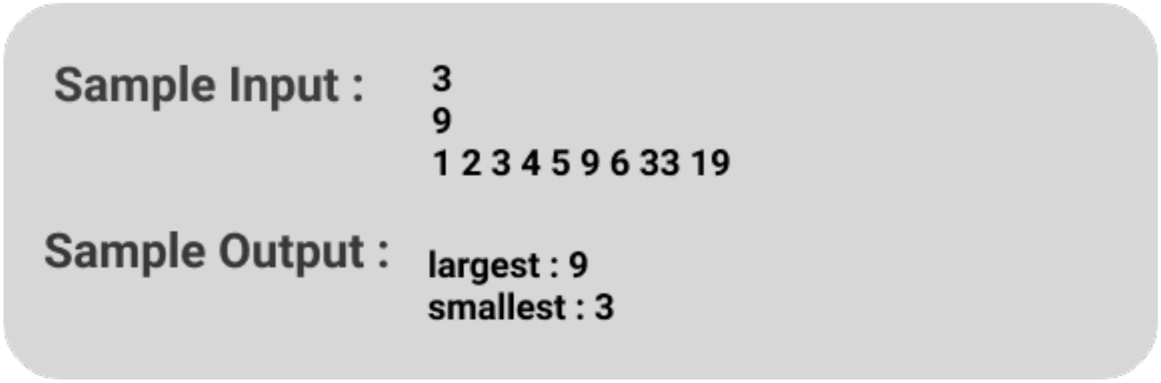
Here, 6 is the size of array and then the elements are input by the user

```
Sample Input :     6
                   -1 0 3 57 89 9

Sample Output :    -1 0 3 9 57 89
```

6) Find the Kth largest and Kth smallest number in an array.

Here , K = 3 , 9 is the size of array :



Sample Input :

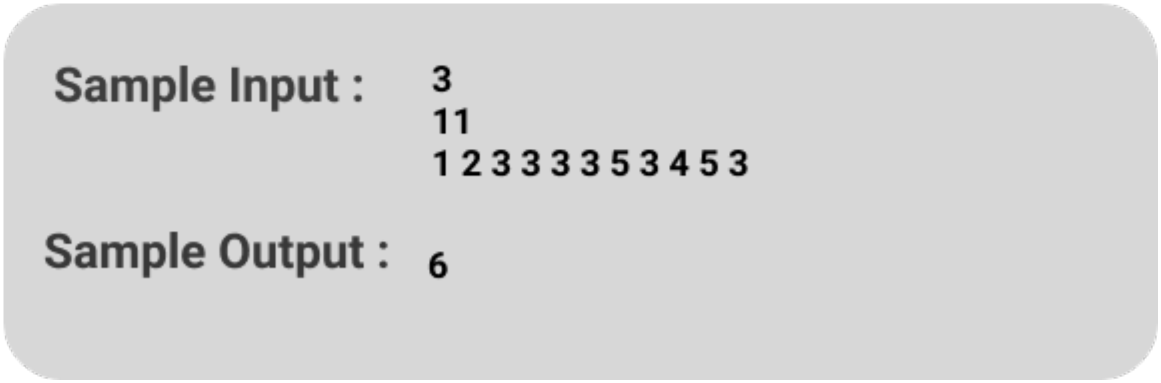
```
3
9
1 2 3 4 5 9 6 33 19
```

Sample Output :

```
largest : 9
smallest : 3
```

7) Given an number n. Find the number of occurrences of n in the array.

Here n = 3 in the sample input , 11 is size of array and occurrence of 3 is 6 times in the given array



Sample Input :

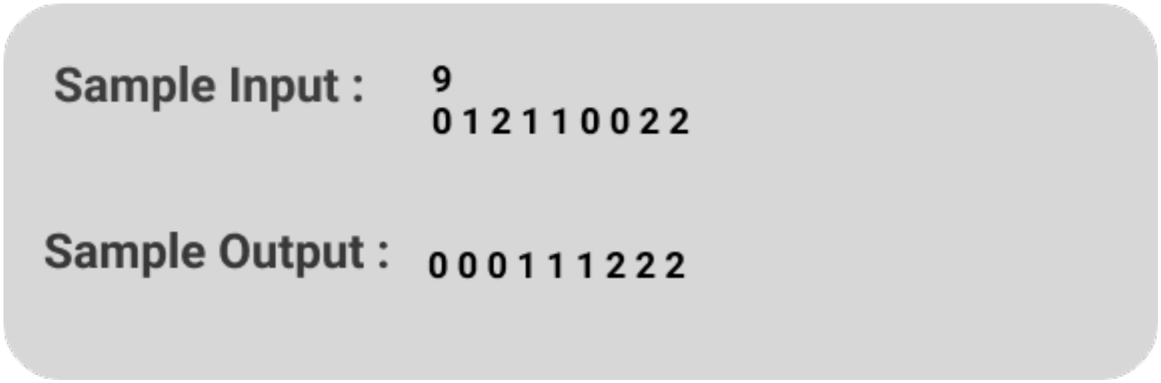
```
3
11
1 2 3 3 3 3 5 3 4 5 3
```

Sample Output :

```
6
```

8) Given an array which consists of only 0, 1 and 2. Sort the array without using any sorting algorithm.

Here, 9 is the size of array input by the user followed by the elements input



Sample Input :

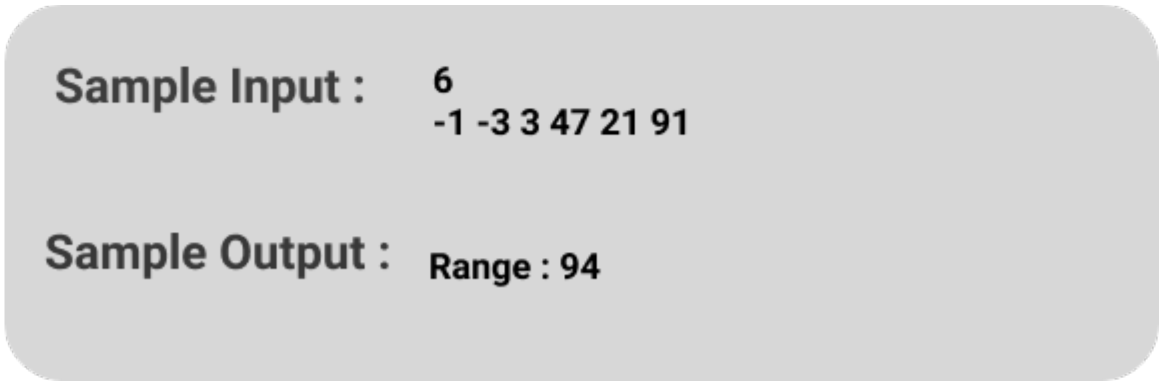
```
9
0 1 2 1 1 0 0 2 2
```

Sample Output :

```
0 0 0 1 1 1 2 2 2
```

9) Find the range of the array. Range means the difference between the maximum and minimum element in the array.

here , 6 is the size of array followed by the input of elements



Sample Input :

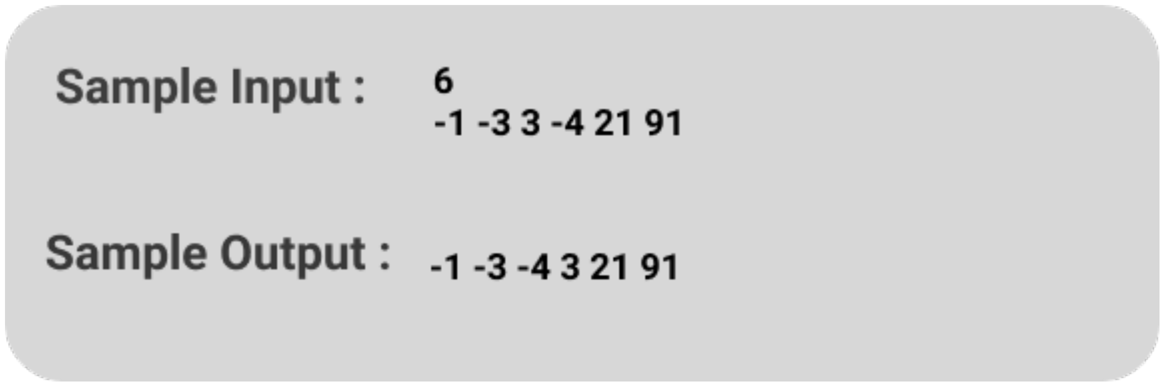
```
6
-1 -3 3 47 21 91
```

Sample Output :

```
Range : 94
```

Here , Range = 91-(-3) = 94

10) Move all the negative elements to one side of the array.



Sample Input :

```
6
-1 -3 3 -4 21 91
```

Sample Output :

```
-1 -3 -4 3 21 91
```

NOTE:
You have to write the code for each question
Such that,
First you have to write the code to input the size of array
Second input the elements also by the user
And then perform the mentioned algorithm

Array

Intermediate level Questions:

1. **Find the Union and Intersection of the two sorted arrays.**
   [practice here: https://practice.geeksforgeeks.org/problems/union-of-two-arrays/0]

2. **Write a program to cyclically rotate an array by one.**
   [Practice here: https://practice.geeksforgeeks.org/problems/cyclically-rotate-an-array-by-one/0]

3. **You are given a list of n-1 integers and these integers are in the range of 1 to n. There are no duplicates in the list. One of the integers is missing in the list. Write an efficient code to find the missing integer.**
   [Practice here: https://practice.geeksforgeeks.org/problems/missing-number-in-array/0]

4. **Find all pairs on integer array whose sum is equal to given number.**
   [Practice here: https://practice.geeksforgeeks.org/problems/count-pairs-with-given-sum/0]

5. **Find duplicates in an array.**
   [Practice here : https://practice.geeksforgeeks.org/problems/find-duplicates-in-an-array/1]

6. **Sort an Array using Quicksort algorithm.**
   [Follow link: https://www.geeksforgeeks.org/quick-sort/]

7. **Find common elements in three sorted arrays**
   [Practice here: https://practice.geeksforgeeks.org/problems/common-elements/0]

8. **Find the first repeating element in an array of integers.**
   [Practice here: https://practice.geeksforgeeks.org/problems/first-repeating-element/0]

9. **Find the first non-repeating element in a given array of integers.**
   [Solution: https://www.geeksforgeeks.org/non-repeating-element/]

10.Given an array with all distinct elements, find the largest three elements. Expected time complexity is O(n) and extra space is O(1).

```
Input: arr[] = {10, 4, 3, 50, 23, 90}
Output: 90, 50, 23
```

11.**Rearrange the array in alternating positive and negative items with O(1) extra space.** [follow link : https://www.geeksforgeeks.org/rearrange-array-alternating-positive-negative-items-o1-extra-space/]

12. Find if there is any subarray with sum equal to zer
[Practice here: https://practice.geeksforgeeks.org/problems/subarray-with-0-sum/0]

13. **Find Largest sum contiguous Subarray.[Very Important]**
[Practice here: https://practice.geeksforgeeks.org/problems/kadanes-algorithm/0]

14. Find the factorial of a large number.
[Practice here: https://practice.geeksforgeeks.org/problems/factorials-of-large-numbers/0]

15. Find Maximum Product Subarray.
[Practice here: https://practice.geeksforgeeks.org/problems/maximum-product-subarray/0]

16. Find longest consecutive subsequence.
[Practice here: https://practice.geeksforgeeks.org/problems/longest-consecutive-subsequence/0]

17. Find the minimum element in a rotated and sorted array.
[Practice here: https://practice.geeksforgeeks.org/problems/minimum-element-in-a-sorted-and-rotated-array/0]

18. **Given an array of size n and a number k, fin all elements that appear more than n/k times.**
[Practice here: https://practice.geeksforgeeks.org/problems/count-element-occurences/1]

19. GCD of given index ranges in an array
[Solution : https://www.geeksforgeeks.org/gcds-of-a-given-index-ranges-in-an-array/]

20. Maximum profit by buying and selling a share at most twice.
[Practice here : https://www.geeksforgeeks.org/maximum-profit-by-buying-and-selling-a-share-at-most-twice/]

21. **Minimize the maximum difference between the heights.** [***ADOBE spl.***]
[Practice here: https://practice.geeksforgeeks.org/problems/minimize-the-heights/0]

22. **Minimum number of Jumps to reach end.**
[Practice here: https://practice.geeksforgeeks.org/problems/minimum-number-of-jumps/0]

23. **Find the two repetitive elements in a given array.**
[Practice here: https://practice.geeksforgeeks.org/problems/two-repeated-elements/0]

24. **Find a triplet that sum to a given value.**
[Practice here: https://practice.geeksforgeeks.org/problems/triplet-sum-in-array/0]

25. **Create an N*M matrix and take input from the user to populate it and then print the matrix**

26. **Find the row with maximum number of 1's.**
[Practice here: https://practice.geeksforgeeks.org/problems/row-with-max-1s/0]

27. **Find the median in a row wise sorted matrix.**
[Practice here: https://practice.geeksforgeeks.org/problems/median-in-a-row-wise-sorted-matrix/0]

28. **Print the matrix in a Spiral manner.** [*Very IMP*]
[Practice here: https://practice.geeksforgeeks.org/problems/spirally-traversing-a-matrix/0]

29. **Find whether an array is a subset of another array.**
[Practice here: https://practice.geeksforgeeks.org/problems/array-subset-of-another-array/0]

30. **Implement two Stacks in an array.**
[Practice here: https://practice.geeksforgeeks.org/problems/implement-two-stacks-in-an-array/1]

STRING

1. **Write a basic program to take input (String) from User and just print it.**

2. **Write a program to count the number of occurrences of each character in the string and print it.**
   [Solution: https://www.geeksforgeeks.org/java-program-count-occurrences-character/]

3. **Write a program to remove all whitespaces in a given string.**
   [Solution: https://www.geeksforgeeks.org/how-to-remove-all-white-spaces-from-a-string-in-java/]

4. **Find Duplicate characters in a string.**
   [Solution: https://www.w3schools.in/java-program/java-program-find-duplicate-characters-string/]

5. **Write a program to reverse the string in place.**
   [Solution: https://www.java67.com/2016/06/how-to-reverse-string-in-place-in-java.html]

6. **Write a program to check whether given two strings are anagram or not.**
   [Practice here: https://practice.geeksforgeeks.org/problems/anagram/0]

7. **Why strings are immutable in Java?**
   [Solution: https://www.geeksforgeeks.org/java-string-is-immutable-what-exactly-is-the-meaning/]

8. **How do you convert string to integer and integer to string in java?**
   [Solution: https://javaconceptoftheday.com/string-to-integer-integer-to-string-conversion-in-java/]

9. **Write a program to reverse each word in the given string.**
   [Solution: https://www.geeksforgeeks.org/reverse-individual-words/]

10. **Check whether the String is a palindrome or not.**
   [Solution: https://practice.geeksforgeeks.org/problems/palindrome-string/0]

String

Intermediate Level Questions:

1. **Write a Code to check whether one string is a rotation of another**
   [Practice here: https://www.geeksforgeeks.org/a-program-to-check-if-strings-are-rotations-of-each-other/]

2. **Write a program to remove Duplicate characters from the String.**
   [Follow here: https://www.geeksforgeeks.org/remove-duplicates-from-a-given-string/]

3. **Write a Program to check whether a string is a valid shuffle of two strings or not.**
   [Follow here: https://www.geeksforgeeks.org/check-whether-a-given-string-is-an-interleaving-of-two-other-given-strings/]

4. **Write a program to find the longest Palindrome in a string.[Lonest palindromic Substring]**
   [Practice here: https://practice.geeksforgeeks.org/problems/longest-palindrome-in-a-string/0]

5. **Find Longest Recurring Subsequence in String.**
   [Practice here: https://practice.geeksforgeeks.org/problems/longest-repeating-subsequence/0]

6. **Print all Subsequences of a string.**
   [Follow here: https://www.geeksforgeeks.org/print-subsequences-string/]

7. **Print all the permutations of the given string**
   [Practice here: https://practice.geeksforgeeks.org/problems/permutations-of-a-given-string/0]

8. **Split the Binary string into two substring with equal 0's and 1's.**
   [Follow here: https://www.geeksforgeeks.org/split-the-binary-string-into-substrings-with-equal-number-of-0s-and-1s/]

9. **Rearrange characters in a string such that no two adjacent are same**
   [Practice here: https://practice.geeksforgeeks.org/problems/rearrange- characters/0]

10. **Write a program to find the smallest window that contains all characters of string itself.**
[Practice here: https://practice.geeksforgeeks.org/problems/smallest-distant-window/0]

11. **Number of Substrings with count of each character as "K".**

12. **Find the longest common subsequence between two strings.**
[Practice here: https://practice.geeksforgeeks.org/problems/longest-common-subsequence/0]

13. **Word Wrap Problem [VERY IMP].**
[Practice here: https://practice.geeksforgeeks.org/problems/word-wrap/0]

14. **Program to generate all possible valid IP addresses from given string.**
[Follow here: https://www.geeksforgeeks.org/program-generate-possible-valid-ip-addresses-given-string/]

15. **EDIT Distance [Very Imp]**
[Practice here: https://practice.geeksforgeeks.org/problems/edit-distance/0]

16. **Find next greater number with same set of digits. [Very Very IMP]**
[Practice here: https://practice.geeksforgeeks.org/problems/next-permutation/0]

17. **Try your hands on all these conversions:**
➔ Prefix to Infix
➔ Prefix to Postfix
➔ Postfix to prefix
➔ Postfix to infix
[Follow link: https://www.geeksforgeeks.org/prefix-infix-conversion/]

18. **Convert a Sentence into its equivalent mobile numeric keypad sequence.**
[Follow here: https://www.geeksforgeeks.org/convert-sentence-equivalent-mobile-numeric-keypad-sequence/]

19. **Balanced Parenthesis problem.[Imp]**
[Practice here: https://practice.geeksforgeeks.org/problems/parenthesis-checker/0]

20. **Minimum number of swaps for bracket balancing.**
[Practice here: https://practice.geeksforgeeks.org/problems/minimum-swaps-for-bracket-balancing/0]

21. **Minimum number of bracket reversals needed to make an expression balanced.**

[Practice here: https://practice.geeksforgeeks.org/problems/count-the-reversals/0]

22. **Word break Problem[Very Imp]**

[Practice here: https://practice.geeksforgeeks.org/problems/word-break/0]

23. **Minimum rotations required to get the same string.**

[Follow here: https://www.geeksforgeeks.org/minimum-rotations-required-get-string/]

24. **Find the first repeated word in string.**

[Practice here: https://practice.geeksforgeeks.org/problems/second-most-repeated-string-in-a-sequence/0]

25. **Efficiently find first repeated character in a string without using any additional data structure in one traversal**

[Practice here : https://practice.geeksforgeeks.org/problems/find-first-repeated-character/0]

26. **Count All Palindromic Subsequence in a given String.**

[Practice here: https://practice.geeksforgeeks.org/problems/count-palindromic-subsequences/1]

27. **Number of flips to make binary string alternate**

[Practice here: https://practice.geeksforgeeks.org/problems/min-number-of-flips/0]

28. **Count of number of given string in 2D character array**

[Follow here: https://www.geeksforgeeks.org/find-count-number-given-string-present-2d-character-array/]

29. Search a Word in a 2D Grid of characters.

[Practice here: https://practice.geeksforgeeks.org/problems/find-the-string-in-grid/0]

30. Boyer Moore Algorithm for Pattern Searching.

[Follow here: https://www.geeksforgeeks.org/boyer-moore-algorithm-for-pattern-searching/]

Stack, Queue and Heap

Basic Level Questions:

Stack:

- Implement a Stack Class with the following methods:
    - push()
    - pop()
    - peek()
    - empty()
    - search()

[Follow here: https://www.geeksforgeeks.org/stack-data-structure-introduction-program/]

- Reverse a String using Stack

[Follow here: https://www.geeksforgeeks.org/stack-set-3-reverse-string-using-stack/]

- Check the expression has valid or Balanced parenthesis or not.

[Follow here: https://www.geeksforgeeks.org/check-for-balanced-parentheses-in-an-expression/]

- Implement two Stacks in an array

[Follow here: https://www.geeksforgeeks.org/implement-two-stacks-in-an-array/]

Queue:

- Implement a Queue class(using arrays) with the following methods:
    - enqueue()
    - dequeue()
    - front()
    - display()

[Follow here: https://www.geeksforgeeks.org/array-implementation-of-queue-simple/]

- Implement a Circular queue with the same methods in the above problem statement

[Follow here: https://www.geeksforgeeks.org/circular-queue-set-1-introduction-array-implementation/]

- Implement a Deque (Doubly Ended Queue) with insertion and deletion allowed at both the ends.

[Follow here: https://www.geeksforgeeks.org/deque-set-1-introduction-applications/]

Heap:

- Implement a Maxheap using arrays and recursion.

[Follow here: https://www.geeksforgeeks.org/max-heap-in-java/]

- Implement a Minheap using arrays and recursion.

[Follow here: https://www.geeksforgeeks.org/min-heap-in-java/]

- Sort an Array using heap. (HeapSort)

[Follow here: https://www.geeksforgeeks.org/cpp-program-for-heap-sort/]

- Maximum of all subarrays of size k.

[Follow here: https://www.geeksforgeeks.org/sliding-window-maximum-maximum-of-all-subarrays-of-size-k/]

Stack, Queue and Heap

Intermediate Level Questions:

Stack:

- Implement Stack using Queues

[Practice here: https://practice.geeksforgeeks.org/problems/stack-using-two-queues/1]

- How to efficiently implement "k" stacks in an array ?

[Follow here: https://www.geeksforgeeks.org/efficiently-implement-k-stacks-single-array/]

- Design a Stack that supports getMin() in O(1) time and O(1) extra space.

[Follow here: https://www.geeksforgeeks.org/design-a-stack-that-supports-getmin-in-o1-time-and-o1-extra-space/]

- Implement stack and Queue using deque

[Follow here: https://www.geeksforgeeks.org/implement-stack-queue-using-deque/]

- Implement methods for Infix to Postfix, Prefix to Infix, Prefix to Postfix, Postfix to Infix and Postfix to prefix Conversion using stack.

[Follow here: https://www.geeksforgeeks.org/stack-set-2-infix-to-postfix/]
[Follow here: https://www.geeksforgeeks.org/prefix-infix-conversion/]
[Follow here: https://www.geeksforgeeks.org/prefix-postfix-conversion/]
[Follow here: https://www.geeksforgeeks.org/postfix-prefix-conversion/]
[Follow here: https://www.geeksforgeeks.org/postfix-to-infix/]

- Find the next Greater element

[Practice here: https://practice.geeksforgeeks.org/problems/next-larger-element/0]

- The celebrity Problem

[Practice here: https://practice.geeksforgeeks.org/problems/the-celebrity-problem/1]

- Arithmetic Expression evaluation

[Practice here: https://www.geeksforgeeks.org/arithmetic-expression-evalution/]

- Evaluation of Postfix expression

[Practice here: https://practice.geeksforgeeks.org/problems/evaluation-of-postfix-expression/0]

- Implement a method to insert an element at its bottom without using any other data structure.

- Reverse a stack using recursion

[Follow here: https://www.geeksforgeeks.org/reverse-a-stack-using-recursion/]

- Sort a Stack using recursion

[Practice here: https://practice.geeksforgeeks.org/problems/sort-a-stack/1]

- Merge Overlapping Intervals

[Practice here: https://practice.geeksforgeeks.org/problems/overlapping-intervals/0]

- Largest rectangular Area in Histogram

[Practice here: https://practice.geeksforgeeks.org/problems/maximum-rectangular-area-in-a-histogram/0]

- Length of the Longest Valid Substring

[Practice here: https://practice.geeksforgeeks.org/problems/valid-substring/0]

- Expression contains redundant bracket or not

[Follow here: https://www.geeksforgeeks.org/expression-contains-redundant-bracket-not/]

- Find the maximum difference between nearest left and right smaller elements

[Practice here: https://practice.geeksforgeeks.org/problems/maximum-difference/1]

- Remove brackets from an algebraic string containing + and – operators

[Follow here: https://www.geeksforgeeks.org/remove-brackets-algebraic-string-containing-operators/]

- Implement a Simple text Editor using Stack

[Follow here: http://algorithmsforgeeks.blogspot.com/2017/03/implement-text-editor-using-stack.html]

- Minimum number of bracket reversals needed to make an expression balanced

[Practice here: https://practice.geeksforgeeks.org/problems/count-the-reversals/0]

Queue:

- Implement Queue using Stack

[Practice here: https://practice.geeksforgeeks.org/problems/queue-using-two-stacks/1]

- LRU Cache Implementation

[Practice here: https://practice.geeksforgeeks.org/problems/lru-cache/1]

- How to efficiently implement "k" queues in an array ?

[Follow here: https://www.geeksforgeeks.org/efficiently-implement-k-queues-single-array/]

- Check if a queue an be sorted into another queue using a stack

[Practice here: https://www.geeksforgeeks.org/check-queue-can-sorted-another-queue-using-stack/]

- Level Order Tree traversal

[Practice here: https://practice.geeksforgeeks.org/problems/level-order-traversal/1]

- Reverse a Queue using recursion

[Practice here: https://practice.geeksforgeeks.org/problems/queue-reversal/1]

- Reverse the first "K" elements of a queue

[Practice here: https://practice.geeksforgeeks.org/problems/reverse-first-k-elements-of-queue/1]

- Interleave the first half of the queue with second half

[Practice here: https://www.geeksforgeeks.org/interleave-first-half-queue-second-half/]

- Sorting a queue without extra space

[Practice here: https://www.geeksforgeeks.org/sorting-queue-without-extra-space/]

- Find the first circular tour that visits all Petrol Pumps

[Practice here: https://practice.geeksforgeeks.org/problems/circular-tour/1]

- Minimum time required to rot all oranges

[Practice here: https://practice.geeksforgeeks.org/problems/rotten-oranges/0]

- Find maximum level sum in Binary tree

[Practice here: https://practice.geeksforgeeks.org/problems/max-level-sum-in-binary-tree/1]

- Distance of nearest cell having 1 in a binary matrix

[Practice here: https://practice.geeksforgeeks.org/problems/distance-of-nearest-cell-having-1/0]

- First negative integer in every window of size "k"

[Practice here: https://practice.geeksforgeeks.org/problems/first-negative-integer-in-every-window-of-size-k/0]

- Check if all levels of two trees are anagrams or not.

[Practice here: https://www.geeksforgeeks.org/check-if-all-levels-of-two-trees-are-anagrams-or-not/]

- Sum of minimum and maximum elements of all subarrays of size "k".

[Practice here: https://www.geeksforgeeks.org/sum-minimum-maximum-elements-subarrays-size-k/]

- Minimum sum of squares of character counts in a given string after removing "k" characters.

[Practice here: https://practice.geeksforgeeks.org/problems/game-with-string/0]

- Queue based approach or first non-repeating character in a stream.

[Practice here: https://practice.geeksforgeeks.org/problems/first-non-repeating-character-in-a-stream/0]

Heap:

- Heap Sort

[Follow here: https://www.geeksforgeeks.org/heap-sort/]

- "k" largest element in an array

[Practice here: https://practice.geeksforgeeks.org/problems/k-largest-elements/0]

- Kth smallest and largest element in an unsorted array

[Practice here: https://practice.geeksforgeeks.org/problems/kth-smallest-element/0]

- Check if a Binary Tree is Heap

[Practice here: https://practice.geeksforgeeks.org/problems/is-binary-tree-heap/1]

- Connect "n" ropes with minimum cost

[Practice here: https://practice.geeksforgeeks.org/problems/minimum-cost-of-ropes/0]

- Merge "K" sorted arrays.

[Practice here: https://practice.geeksforgeeks.org/problems/merge-k-sorted-arrays/1]

- Largest Derangement of a Sequence

[Practice here: https://www.geeksforgeeks.org/largest-derangement-sequence/]

- Maximum distinct elements after removing "k" elements

[Practice here: https://practice.geeksforgeeks.org/problems/maximum-distinct-elements-after-removing-k-elements/0]

- Median in a stream of Running Integers

[Practice here: https://practice.geeksforgeeks.org/problems/find-median-in-a-stream/0]

- Largest Triplet Product in a stream

[Practice here: https://www.geeksforgeeks.org/largest-triplet-product-stream/]

- Convert BST to Min Heap

[Practice here: https://www.geeksforgeeks.org/convert-bst-min-heap/]

- Merge 2 Binary Max Heaps

[Practice here: https://practice.geeksforgeeks.org/problems/merge-two-binary-max-heap/0]

- Kth largest sum continuous subarrays

[Practice here: https://www.geeksforgeeks.org/k-th-largest-sum-contiguous-subarray/]

- Convert min heap to max heap

[Practice here: https://www.geeksforgeeks.org/convert-min-heap-to-max-heap/]

- Why is Binary Heap is preferred over BST for Priority Queue ?

[Answer: https://www.geeksforgeeks.org/why-is-binary-heap-preferred-over-bst-for-priority-queue/]

- Given Level order traversal of a Binary Tree, check if the tree is Min heap.

[Follow here: https://www.geeksforgeeks.org/given-level-order-traversal-binary-tree-check-tree-min-heap/]

- Rearrange characters in a string such that no two adjacent are same.

[Practice here: https://practice.geeksforgeeks.org/problems/rearrange-characters/0]

- Minimum sum of two numbers formed from digits of an array

[Practice here: https://practice.geeksforgeeks.org/problems/min-sum-formed-by-digits/0]

- Leetcode- reorganize strings

[Practice here: https://leetcode.com/problems/reorganize-string/]

- Merge "K" Sorted Linked Lists

[Practice here: https://practice.geeksforgeeks.org/problems/merge-k-sorted-linked-lists/1]

- Smallest range in "K" Lists

[Practice here: https://practice.geeksforgeeks.org/problems/find-smallest-range-containing-elements-from-k-lists/1]

Linked List

Basic Level Questions:

1. Create a Singly Linked list class (members-> value and next pointer), with the following methods:
   - ➔createNewNode()
   - ➔addNodeAtBegin()
   - ➔addNodeAtEnd()
   - ➔length()
   - ➔print()

   [Follow: https://www.geeksforgeeks.org/linked-list-set-1-introduction/]

2. Create a method to search an element in the above created linked list

3. Create a method to delete any Element in the above create linked list

4. Create a method to provide the "Nth" Node from the above created linked list.

5. Create a method to Count the numbers of a specific value in the above created linked list

6. Find the minimum and maximum element in the linked list.

7. Convert the above created linked list into a Circular Linked List.

8. Create a Doubly Linked list and perform all the operations that were done on the above singly linked list.
   [Follow : https://www.geeksforgeeks.org/doubly-linked-list/]

9. After performing all the above operations convert the above created DLL into a Circular doubly Linked list.

Linked List

Intermediate Level Questions:

1. Write a program to get the “N th” Node from the end of the Singly Linked List.
   [Practice here: https://practice.geeksforgeeks.org/problems/nth-node-from-end-of-linked-list/1]

2. Write a Program to check whether the Singly Linked list is a palindrome or not.
   [Practice here: https://practice.geeksforgeeks.org/problems/check-if-linked-list-is-pallindrome/1]

3. Write a Program to reverse the Linked List. (Both Iterative and recursive)
   [Practice here: https://practice.geeksforgeeks.org/problems/reverse-a-linked-list/1]

4. Reverse a Linked List in group of Given Size. [**Very Imp**]
   [Practice here: https://practice.geeksforgeeks.org/problems/reverse-a-linked-list-in-groups-of-given-size/1]

5. Write a program to Detect loop in a linked list.
   [Practice here: https://practice.geeksforgeeks.org/problems/detect-loop-in-linked-list/1]

6. Write a program to find the length of loop in the linked list.
   [Practice here: https://practice.geeksforgeeks.org/problems/find-length-of-loop/1]

7. Write a function to delete the Linked List.
   [Follow: https://www.geeksforgeeks.org/write-a-function-to-delete-a-linked-list/]

8. Remove Duplicates in a sorted Linked List.
   [Practice here: https://practice.geeksforgeeks.org/problems/remove-duplicate-element-from-sorted-linked-list/1]

9. Remove Duplicates in a Unsorted Linked List.
   [Practice here: https://practice.geeksforgeeks.org/problems/remove-duplicates-from-an-unsorted-linked-list/1]

10. Write a Program to Move the last element to Front in a Linked List.
[Follow: https://www.geeksforgeeks.org/move-last-element-to-front-of-a-given-linked-list/]

11. Add "1" to a number represented as a Linked List.
[Practice here: https://practice.geeksforgeeks.org/problems/add-1-to-a-number-represented-as-linked-list/1]

12. Add two numbers represented by linked lists.
[Practice here: https://practice.geeksforgeeks.org/problems/add-two-numbers-represented-by-linked-lists/1]

13. Intersection of two Sorted Linked List.
[Practice here:https://practice.geeksforgeeks.org/problems/intersection-of-two-sorted-linked-lists/1]

14. Intersection Point of two Linked Lists.
[Practice here: https://practice.geeksforgeeks.org/problems/intersection-point-in-y-shapped-linked-lists/1]

15. Merge Sort For Linked lists.[**Very Important**]
[Follow: https://www.geeksforgeeks.org/merge-sort-for-linked-list/]

16. Quicksort for Linked Lists.[**Very Important**]
[Follow: https://www.geeksforgeeks.org/quicksort-on-singly-linked-list/]

17. Find the middle Element of a linked list.
[Practice here: https://practice.geeksforgeeks.org/problems/finding-middle-element-in-a-linked-list/1]

18. Check if a linked list is a circular linked list.
[Practice here: https://practice.geeksforgeeks.org/problems/circular-linked-list/1]

19. Split a Circular linked list into two halves.
[Practice here: https://practice.geeksforgeeks.org/problems/split-a-circular-linked-list-into-two-halves/1]

20. Deletion from a Circular Linked List.
[Follow here: https://www.geeksforgeeks.org/deletion-circular-linked-list/]

21. Count Nodes in a Circular Linked List.
[Follow here: https://www.geeksforgeeks.org/count-nodes-circular-linked-list/]

22. Exchange first and last nodes in a linked list.
[Follow here: https://www.geeksforgeeks.org/exchange-first-last-node-circular-linked-list/]

23. Reverse a Doubly Linked list.
[Practice here: https://practice.geeksforgeeks.org/problems/reverse-a-doubly-linked-list/1]

24. Find pairs with a given sum in a DLL.
[Follow here: https://www.geeksforgeeks.org/find-pairs-given-sum-doubly-linked-list/]

25. Count triplets in a sorted DLL whose sum is equal to given value "X".
[Follow here: https://www.geeksforgeeks.org/count-triplets-sorted-doubly-linked-list-whose-sum-equal-given-value-x/]

26. Sort a "k" sorted Doubly Linked list.[**Very IMP**]
[Follow here: https://www.geeksforgeeks.org/sort-k-sorted-doubly-linked-list/]

27. Rotate Doubly Linked list by N nodes.
[Follow here: https://www.geeksforgeeks.org/rotate-doubly-linked-list-n-nodes/]

28. Rotate a Doubly Linked list in group of Given Size.[**Very IMP**]
[Follow here: https://www.geeksforgeeks.org/reverse-doubly-linked-list-groups-given-size/]

29. Can we reverse a linked list in less than O(n) ?
[Study : https://www.geeksforgeeks.org/can-we-reverse-a-linked-list-in-less-than-on/]

30. Why Quicksort is preferred for. Arrays and Merge Sort for Linked Lists ?
[Study : https://www.geeksforgeeks.org/why-quick-sort-preferred-for-arrays-and-merge-sort-for-linked-lists/]

TREE

Basic Level Questions:

- Create a class Tree consisting of 3 members (data, left pointer and right pointer) , including all these functions:
    - Insertion of Node
    - Deletion of Node
    - Inorder Traversal (Recursive and Iterative)
    - Preorder Traversal (Recursive and Iterative)
    - Postorder Traversal (Recursive and Iterative)
    - Level Order Traversal
    - Reverse Level Order traversal
    - Searching of Value
    - Height of tree
    - Diameter of Tree
    - Mirror of Tree
    - Check tree is balanced or not
    - Find minimum value in tree
    - Find maximum value in tree

[Follow here: https://www.geeksforgeeks.org/binary-tree-data-structure/]

- Create a class BST(Binary Search Tree) consisting of 3 members (data, left pointer and right pointer) , including all these functions:
    - Insertion into BST
    - Deletion from BST
    - Level order print
    - Traversal(inorder , preorder and postorder)
    - Searching a value in BST
    - Check if is BST or not
    - Find inorder successor and inorder predecessor
    - Print all root node to leaf node paths
    - Find min and max value in BST

[Follow here: https://www.geeksforgeeks.org/binary-search-tree-data-structure/]

TREE

Intermediate Level Questions:

Binary Tree:

- Print top view, bottom view , left view and right view of a binary tree.

[Follow here: https://www.geeksforgeeks.org/print-nodes-top-view-binary-tree/]
[Practice here: https://practice.geeksforgeeks.org/problems/top-view-of-binary-tree/1]
[Practice here: https://practice.geeksforgeeks.org/problems/bottom-view-of-binary-tree/1]
[Practice here: https://practice.geeksforgeeks.org/problems/left-view-of-binary-tree/1]
[Practice here: https://practice.geeksforgeeks.org/problems/right-view-of-binary-tree/1]

- Find Nth node of Inorder Traversal

[Follow here: https://www.geeksforgeeks.org/find-n-th-node-inorder-traversal/]

- Print Level Order Traversal in Spiral Form

[Practice here: https://practice.geeksforgeeks.org/problems/level-order-traversal-in-spiral-form/1]

- Print Diagonal Traversal of a Binary Tree

[Practice here: https://practice.geeksforgeeks.org/problems/diagonal-traversal-of-binary-tree/1]

- Print Boundary Traversal of Binary Tree

[Practice here: https://practice.geeksforgeeks.org/problems/boundary-traversal-of-binary-tree/1]

- Construct a Binary Tree from given Inorder and Preorder traversal

[Practice here: https://practice.geeksforgeeks.org/problems/construct-tree-1/1]

- Construct a Binary Tree from Inorder and Level order traversal

[Practice here: https://practice.geeksforgeeks.org/problems/construct-tree-from-inorder-and-levelorder/1]

- Construct Binary Tree from String with Bracket Representation

[Follow here: https://www.geeksforgeeks.org/construct-binary-tree-string-bracket-representation/]

- Convert a Binary Tree into Doubly Linked List(DLL)

[Practice here: https://practice.geeksforgeeks.org/problems/binary-tree-to-dll/1]

- Convert a Given Binary Tree into a Sum Tree

[Practice here: https://practice.geeksforgeeks.org/problems/transform-to-sum-tree/1]

- Find minimum swaps required to convert a Binary tree into Binary Search Tree

[Follow here: https://www.geeksforgeeks.org/minimum-swap-required-convert-binary-tree-binary-search-tree/]

- Check if Binary Tree is Sum tree or not

[Practice here: https://practice.geeksforgeeks.org/problems/sum-tree/1]

- Check if All leaf node are at same level or not

[Practice here: https://practice.geeksforgeeks.org/problems/leaf-at-same-level/1]

- Check if a Binary Tree contains duplicate subtrees of size 2 or more.

[Practice here: https://practice.geeksforgeeks.org/problems/duplicate-subtree-in-binary-tree/1]

- Check if two trees are mirror

[Practice here: https://practice.geeksforgeeks.org/problems/check-mirror-in-n-ary-tree/0]

- Check if given graph is tree or not

[Follow here: https://www.geeksforgeeks.org/check-given-graph-tree/]

- Sum of Nodes on the longest path from root to leaf node

[Practice here: https://practice.geeksforgeeks.org/problems/sum-of-the-longest-bloodline-of-a-tree/1]

- Find Largest subtree sum in a tree

[Follow here: https://www.geeksforgeeks.org/find-largest-subtree-sum-tree/]

- Maximum sum of nodes in Binary Tree such that no two are adjacent

[Practice here: https://www.geeksforgeeks.org/maximum-sum-nodes-binary-tree-no-two-adjacent/]

- Print all k-sum paths in a Binary Tree

[Practice here: https://practice.geeksforgeeks.org/problems/k-sum-paths/1]

- Find Lowest Common Ancestor in a Binary Tree

[Practice here: https://practice.geeksforgeeks.org/problems/lowest-common-ancestor-in-a-binary-tree/1]

- Find distance between two nodes in a Binary Tree

[Practice here: https://practice.geeksforgeeks.org/problems/min-distance-between-two-given-nodes-of-a-binary-tree/1]

- Kth Ancestor of a node in a Binary tree

[Follow here: https://www.geeksforgeeks.org/kth-ancestor-node-binary-tree-set-2/]

- Find All Duplicate Subtrees in a Binary Tree

[Practice here: https://practice.geeksforgeeks.org/problems/duplicate-subtrees/1]

- Tree Isomorphism Problem

[Practice here: https://practice.geeksforgeeks.org/problems/check-if-tree-is-isomorphic/1]

Binary Search Tree:

- Construct BST from inorder and preorder traversal
- Construct BST from inorder and postorder traversal

- Construct BST from Preorder Traversal

[Follow here: https://www.geeksforgeeks.org/construct-bst-from-given-preorder-traversa/]

- Convert Binary Tree into BST

[Practice here: https://practice.geeksforgeeks.org/problems/binary-tree-to-bst/1]

- Convert a normal BST into balanced BST

[Follow here: https://www.geeksforgeeks.org/convert-normal-bst-balanced-bst/]

- Merge two BST [Very Important]

[Practice here: https://practice.geeksforgeeks.org/problems/merge-two-bst-s/1]

- Find Lowest Common Ancestor (LCA) of BST

[Practice here: https://practice.geeksforgeeks.org/problems/lowest-common-ancestor-in-a-bst/1]

- Find Kth Largest Element in a BST

[Practice here: https://practice.geeksforgeeks.org/problems/kth-largest-element-in-bst/1]

- Count pairs from Two BSTs whose sum is equal to given value x.

[Practice here: https://practice.geeksforgeeks.org/problems/brothers-from-different-root/1]

- Find the median of BST in O(n) time and O(1) space
  [Follow here: https://www.geeksforgeeks.org/find-median-bst-time-o1-space/]

- Count BST nodes that lies in the given range
  [Practice here: https://practice.geeksforgeeks.org/problems/count-bst-nodes-that-lie-in-a-given-range/1]

- Replace every element with the least greater element on its right
  [Practice here: https://www.geeksforgeeks.org/replace-every-element-with-the-least-greater-element-on-its-right/]

- Given "n" appointments, find the conflicting appointments
  [Practice here: https://www.geeksforgeeks.org/given-n-appointments-find-conflicting-appointments/]

- Populate inorder successor of all nodes.
  [Practice here: https://practice.geeksforgeeks.org/problems/populate-inorder-successor-for-all-nodes/1]

- Check Dead in a BST
  [Practice here: https://practice.geeksforgeeks.org/problems/check-whether-bst-contains-dead-end/1]

- Check preorder is valid or not
  [Practice here: https://practice.geeksforgeeks.org/problems/preorder-to-postorder/0]

Expression tree:

- Evaluate Expression tree.
  [Practice here: https://practice.geeksforgeeks.org/problems/expression-tree/1]

AVL Tree:

- Insertion and Deletion only
  Follow here:
  [Insertion: https://www.geeksforgeeks.org/avl-tree-set-1-insertion/]
  [Deletion: https://www.geeksforgeeks.org/avl-tree-set-2-deletion/]

RBL Tree:

- Insertion and Deletion only
  Follow here:
  [Intro: https://www.geeksforgeeks.org/red-black-tree-set-1-introduction-2/]
  [Insertion: https://www.geeksforgeeks.org/red-black-tree-set-2-insert/]
  [Deletion: https://www.geeksforgeeks.org/red-black-tree-set-3-delete-2/]

B Tree and B$^+$ Tree:

- Go through theory only

[B tree: https://www.geeksforgeeks.org/introduction-of-b-tree-2/]
[B$^+$ Tree: https://www.geeksforgeeks.org/introduction-of-b-tree/]

Set and Map

Intermediate Level Questions:

- Replace repeating elements with greater than greatest values

  [Follow here: https://www.geeksforgeeks.org/replace-repeating-elements-with-greater-that-greatest-values/]

- Replace duplicate with greater than previous duplicate value

  [Follow here: https://www.geeksforgeeks.org/replace-duplicates-with-greater-than-previous-duplicate-value/]

- Construct a tree from inorder and level order traversals.

  [Follow here: https://www.geeksforgeeks.org/construct-tree-inorder-level-order-traversals-set-2/]

- Print all triplet in sorted array that form AP

  [Follow here: https://www.geeksforgeeks.org/print-triplets-sorted-array-form-ap/]

- Number of Unique triplets whose XOR is Zero

  [Follow here: https://www.geeksforgeeks.org/number-unique-triplets-whose-xor-zero/]

- Find if there is a subarray with sum 0

  [Follow here: https://www.geeksforgeeks.org/find-if-there-is-a-subarray-with-0-sum/]

- Maximize elements using another array

  [Follow here: https://www.geeksforgeeks.org/maximize-elements-using-another-array/]

- Print array elements that are divisible by at least one other

  [Follow here: https://www.geeksforgeeks.org/divisibility-check/]

- Check if a pair with given product exists in Linked List

  [Follow here: https://www.geeksforgeeks.org/check-if-a-pair-with-given-product-exists-in-linked-list/]

- Longest Subarray with only one value greater than “k”

  [Follow here: https://www.geeksforgeeks.org/longest-subarray-in-which-all-elements-are-greater-than-k/]

- Check if a pair with given absolute difference exists in a Matrix

  [Follow here: https://www.geeksforgeeks.org/check-if-a-pair-with-given-absolute-difference-exists-in-a-matrix/]

- Check if a pair with given product exists in a Matrix

  [Follow here: https://www.geeksforgeeks.org/check-if-a-pair-with-given-product-exists-in-a-matrix/]

- Count unique numbers that can be generated from “N” by adding one and removing trailing zeros

  [Follow here: https://www.geeksforgeeks.org/count-unique-numbers-that-can-be-generated-from-n-by-adding-one-and-removing-trailing-zeros/]

- Total distinct pairs of ugly numbers from two arrays

  [Follow here: https://www.geeksforgeeks.org/total-distinct-pairs-of-ugly-numbers-from-two-arrays/]

- Median in a Stream using Set

  [Follow here: https://www.hackerrank.com/challenges/find-the-running-median/problem]

Some more questions are there in the pdf attached with name “Map Questions”. Don’t skip them as they are very important and frequently comes in tech interviews.

Set and Map

Basic Level Questions:

- Implement a class Map using arrays or vectors, which performs the following operation in O(1) Time Complexity:
    - Insert
    - Delete
    - Find
    - GetRandom (gets you any random value from the ones which are present inside the map currently)

- Find the only repetitive number between 1 to n-1
  [Follow here: https://www.geeksforgeeks.org/find-repetitive-element-1-n-1/]

- Difference between set, multiset, unordered_set, unordered_multiset.
  [Follow here: https://www.geeksforgeeks.org/difference-set-multiset-unordered_set-unordered_multiset/]

- Find the only element that appears "b" times
  [Follow here: https://www.geeksforgeeks.org/find-element-appears-b-times/]

- Remove Duplicate or Repeated words from String
  [Follow here: https://www.geeksforgeeks.org/remove-duplicaterepeated-words-string/]

- Find total no. of distinct years from a string
  [Follow here: https://www.geeksforgeeks.org/find-total-number-of-distinct-years-from-a-string/]

- Equally divide into 2 sets such that one set has maximum distinct elements
  [Follow here: https://www.geeksforgeeks.org/equally-divide-into-two-sets-such-that-one-set-has-maximum-distinct-elements/]

- Check if a pair with given product exist in a Linked List
  [Follow here: https://www.geeksforgeeks.org/check-if-a-pair-with-given-product-exists-in-linked-list/]

- Check loop in linked list and remove the loop using map
  [Follow here: https://www.geeksforgeeks.org/detect-and-remove-loop-in-a-linked-list/]

- Count of pairs between 2 arrays such that the sums are distinct
  [Follow here: https://www.geeksforgeeks.org/count-of-pairs-between-two-arrays-such-that-the-sums-are-distinct/]

- Kth missing element in an unsorted array
  [Follow here: https://www.geeksforgeeks.org/k-th-missing-element-in-an-unsorted-array/]

- Number of Strings that satisfy the given condition in the link below
  [Follow here: https://www.geeksforgeeks.org/number-of-strings-that-satisfy-the-given-condition/]

- Number of ways to choose an integer such that there are exactly "k" elements greater than it in the given array
  [Follow here: https://www.geeksforgeeks.org/noble-integers-in-an-array-count-of-greater-elements-is-equal-to-value/]

- Number of unique pairs in an array
  [Follow here: https://www.geeksforgeeks.org/number-of-unique-pairs-in-an-array/]

- Largest Subset possible for an array satisfying the given condition in the link below:
  [Follow here: https://www.geeksforgeeks.org/largest-sub-set-possible-for-an-array-satisfying-the-given-condition/]

- Check if the array has an element which is equal to product of remaining elements
  [Follow here: https://www.geeksforgeeks.org/check-if-the-array-has-an-element-which-is-equal-to-product-of-remaining-elements/]

- Find if array has an element whose value is half of array sum.
  [Follow here: https://www.geeksforgeeks.org/find-if-array-has-an-element-whose-value-is-half-of-array-sum/]