Logical Class Questions

By Amarjeet

Total: 232

Arrays: (76 questions)

Day 1(Basic 15 Questions):

Ques – 1

Write a program to store array dynamically by reading the size and elements from the user.

Input as:

Enter size of the array:6

Enter 6 elements: 1 2 3 4 5 6

Output is:

Array elements are: 1 2 3 4 5 6

Ques – 2

Write a java program to find the middle elements of a given array. If array is odd then print only the middle element, if array is even then print both the middle elements. [searching based on index]

Input as : 1 2 3 4 5

Output is : 3

Input is :1 2 3 4 5 6

Output is :3 4

.

Ques – 3

Write a java program to swap the last and first element of a given array.

[Swapping Operation]

Input as: [1,2,3,4,5,6]

Output as: 6 2 3 4 5 1

Ques – 4

Write a java program to count how many even elements and how many odd elements are present in an array.[Searching Operation]

Input as : 1 2 3 4 5

Ans :

Even count : 2

Odd count : 3

Ques – 5

Write a java program to print all even elements and odd elements separately from an array.[Searching and retrieving]

Input as : 1 2 3 4 5 6 7

Even elements are : 2 4 6

Odd elements are : 1 3 5 7

Q6) Find Maximum Element in an Array.

Input as: 6 3 2 1 5 5 4

Output as: Min is: 1

Q7) Find Minimum element in an array.

Input as: 6 3 2 1 5 8 5 4

Output as: Max is: 8

Q8) Calculate Sum of Array Elements.

Input as: 1 2 3 4 5

Output as: 15

Q9) Print all elements in reverse order of an Array.

```
                0 1 2 3 4
     Input as:  1 2 3 4 5
    Output as:  5 4 3 2 1
```

Q10) Find Average of Array Elements.

Q11) Search for a given Element in an Array if present then print "Element found", if not present print "Element not found"

Input as: 1 4 3 5 2 6 5

Enter searching element: 5

Output: Element found

Input as: 1 4 3 5 2 6

Enter searching element: 7

Output: Element not found

Q12) Search for a given Element in an Array if present print its location.

Q13) Find the median of a sorted array.

```
 Test Case: 1
----------------
 Array size: 6
 Elements are: 1, 2, 3, 4, 5, 6
 Median is: 3.5
Test Case: 2
----------------
 Array size: 5
 Elements are: 1, 2, 3, 4, 5
 Median is: 3
```

Q14) Reverse the array and print it.

Q15) Check the given array is sorted(it can be Ascending order or descending order) or not if sorted then return true if not sorted then return false.

```
int[] a = {13, 4, 23,  2, 2, 4}
Output : Is sorted : false
```

int[] a = {3,5,6,7,8}

Output : Is sorted : true

int[] a = {8,6,5,3,1}

Output: Is sorted : true

int[] a = {3,5,6,7,8,1,3,5}

Output: Is sorted : false

Day 2 (Basic 10 Questions):

Ques - 1

Write a java program to count how many elements are present more than one time.

Input : arr = [1,2,1,3,4,6,2,5,4]

Output : 3 elements are present more than one time

Ques - 2

Write a java program to print all duplicate elements from a given array.

Input: arr = [1,2,1,3,4,6,2,5,4]

Output: 1 2 4

Ques - 3

Write a java program to print all unique elements from a given array.

Input: arr = [1,2,1,3,4,6,2,5,4]

Output: 3 6 5

Ques - 4

Write a Java method named modifyArrayElement that modifies the given array by replacing every occurrence of oldElement with newElement.

arr = [1, 2, 3, 4, 5, 2, 3]

oldElement = 3

newElement = 10

After modification, arr should be [1, 2, 10, 4, 5, 10]

Ques - 5

Write a Java method named replaceElement that replaces the element at the given position in the array with another given element.

Input:

arr = [1, 2, 3, 4, 5]

position = 2

newElement = 10

After modification, arr should be [1, 2, 10, 4, 5]

Ques - 6

Write java method named findEvenandOddPositionElement that print all the elements which is present at even index and also in odd index.

Ques - 7

Write java method named findSumOfOddandEvenPositionElement that print the all odd positions sum and also even

position sum.

Ques - 8

Write java method named findPrimePositionElement that print all the elements which is present at prime index.

Ques - 9

Implement a Java program to find the frequency of each element in an integer array and display the results.

Ques - 10

Implement a java program to find kth largest element in an unsorted array and print it.

Day 3(Basic 10 questions)

Ques – 1

Given a sorted array of integers, remove duplicates such that each element appears only once. Print the elements without duplicates.

Input:

nums = [1, 1, 2, 2, 3, 4, 4, 5]

Output:

[1, 2, 3, 4, 5]

Ques – 2

Write a program to insert one element in a desired index in an array and return resulting array.

arr = {1,2,3,4,5}

Test Case - 1

Enter index: 3

Enter element: 9

Output : [1, 2, 3, 9, 4, 5]

Test Case - 2

```
------------------
Enter index: 0
Enter element: 9
Output : [9, 1, 2, 3, 4, 5]
```

Test Case - 3

```
------------------
Enter index : 4
Enter element : 9
Output : [1, 2, 3, 4, 9, 5]
```

Ques – 3

Write a program to merge two sorted arrays into a single sorted array.

Input:

nums1 = [1, 3, 5]

nums2 = [2, 4, 6]

Output:

[1, 2, 3, 4, 5, 6]

Ques – 4

You are given an array of integers and a number of steps. Implement a method to rotate the array to the right by the given number of steps.

Input:

nums = [1, 2, 3, 4, 5]

steps = 2

Output:

[4, 5, 1, 2, 3]

Input :

nums = [1,2,3,4,5]

steps = 5

Output :

[1,2,3,4,5]

Input:

nums = [1,2,3,4,5]

steps = 7

Output :

[4, 5, 1, 2, 3]

Ques – 5

You are given an array of integers and an index number. Implement a method to rotate left from the given index.

arr = [1,2,3,4,5,6]

int index = 3

Output : 4 5 6 1 2 3

arr = [1,2,3,4,5,6,7,8]

int index = 3

output: 4 5 6 7 8 1 2 3

Ques – 6

You are given a sorted array containing n-1 unique integers in the range from 1 to n. There is exactly one integer missing. Write a method to find and return the missing integer.

Input:

nums = [1, 2, 4, 5, 6]

Output: 3

Ques – 7

You are given a sorted array containing n-1 unique integers in the range from 1 to n. There may be one or more integers missing from the array. Write a method to find and print all missing integers.

Example:

Input:

nums = [1, 2, 4, 7, 19]

Output:

[3, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]

Ques – 8

Write a java program to count and print all the elements which contains 2 in it.

Input as : {1,12,32,44,222,102,34,56,67,125}

Output as : 12, 32, 222, 102, 125

Ques – 9

Write a java program to insert an element in its proper position

in a given sorted array.

Example :

Input as : {5,6,14 ,17,44}

Element to be inserted :15

Result array : 5,6,14,15,17,44

Ques – 10

Write a java program to shift all the zeros to the end of the given array without changing the order of non-zero elements.

Input as : {1,0,0,4,5,0,0,1};

Output as : {1,4,5,1,0,0,0,0}

Ques – 11

Write a java program to shift all the zeros to the front of the given array without changing the order of non-zero elements.

Input as : {1,0,0,4,5,0,0,1};

Output as : {0,0,0,0,1,4,5,1}

Day 4(3 Sorting Algorithm):

Ques – 1

Write a java program to sort the array in ascending and descending order by using Bubble sorting.

Ques – 2

Write a java program to sort the array in ascending and descending order by using Selection sorting.

Ques – 3

Write a java program to sort the array in ascending and descending order by using Insertion sorting.

Day 5(5 Leet Code Questions):

Ques - 1

Find first non-repeating element in a given Array of integers

Given an array of integers of size N, the task is to find the first non-repeating element in this array.

Examples:

Input: {-1, 2, -1, 3}

Output: 2

Explanation: The first number that does not repeat is : 2

Input: {9, 4, 9, 6, 7, 4}

Output: 6

Ques - 2

Given an array having positive and negative numbers, our task is to arrange them in an alternate fashion such that every positive number is followed by a negative number and vice-versa maintaining the order of appearance. The number of positive and negative numbers need not to be equal. If there are more positive numbers then they have to appear at the end of the array, same condition for negative numbers also.

Examples:

Input: arr[] = {1, 2, 3, -4, -1, 4}

Output: arr[] = {-4, 1, -1, 2, 3, 4}

Input: arr[] = {-5, -2, 5, 2, 4, 7, 1, 8, 0, -8}

Output: arr[] = {-5, 5, -2, 2, -8, 4, 7, 1, 8, 0}

Ques - 3

Shuffle the Array:

Given the array nums consisting of 2n elements in the form [x1,x2,...,xn,y1,y2,...,yn].

Return the array in the form [x1,y1,x2,y2,...,xn,yn].

Example 1:

Input: nums = [2,5,1,3,4,7], n = 3

Output: [2,3,5,4,1,7]

Explanation: Since x1=2, x2=5, x3=1, y1=3, y2=4, y3=7 then the answer is [2,3,5,4,1,7].

Example 2:

Input: nums = [1,2,3,4,4,3,2,1], n = 4

Output: [1,4,2,3,3,2,4,1]

Example 3:

Input: nums = [1,1,2,2], n = 2

Output: [1,2,1,2]

Ques - 4

Number of Employees Who Met the Target:

There are n employees in a company, numbered from 0 to n - 1. Each employee i has worked for hours[i] hours in the company.

The company requires each employee to work for at least target hours.

You are given a 0-indexed array of non-negative integers hours of length n and a non-negative integer target.

Return the integer denoting the number of employees who worked at least target hours.

Example 1:

Input: hours = [0,1,2,3,4], target = 2

Output: 3

Explanation: The company wants each employee to work for at least 2 hours.

- Employee 0 worked for 0 hours and didn't meet the target.

- Employee 1 worked for 1 hours and didn't meet the target.

- Employee 2 worked for 2 hours and met the target.

- Employee 3 worked for 3 hours and met the target.

- Employee 4 worked for 4 hours and met the target.

There are 3 employees who met the target.

Example 2:

Input: hours = [5,1,4,2,2], target = 6

Output: 0

Explanation: The company wants each employee to work for at least 6 hours.

There are 0 employees who met the target.

Kids With the Greatest Number of Candies:

There are n kids with candies. You are given an integer array candies, where each candies[i] represents the number of candies the ith kid has, and an integer extraCandies, denoting the number of extra candies that you have.

Return a boolean array result of length n, where result[i] is true if, after giving the ith kid all the extraCandies, they will have the greatest number of candies among all the kids, or false otherwise.

Note that multiple kids can have the greatest number of candies.

Example 1:

Input: candies = [2,3,5,1,3], extraCandies = 3

Output: [true,true,true,false,true]

Explanation: If you give all extraCandies to:

- Kid 1, they will have 2 + 3 = 5 candies, which is the greatest among the kids.
- Kid 2, they will have 3 + 3 = 6 candies, which is the greatest among the kids.
- Kid 3, they will have 5 + 3 = 8 candies, which is the greatest among the kids.
- Kid 4, they will have 1 + 3 = 4 candies, which is not the greatest among the kids.
- Kid 5, they will have 3 + 3 = 6 candies, which is the greatest among the kids.

Example 2:

Input: candies = [4,2,1,1,2], extraCandies = 1

Output: [true,false,false,false,false]

Explanation: There is only 1 extra candy.

Kid 1 will always have the greatest number of candies, even if a different kid is given the extra candy.

Example 3:

Input: candies = [12,1,12], extraCandies = 10

Output: [true,false,true]

Day 6(4 Leet Code Questions):

Ques - 1

Given an integer array nums, return an array answer such that answer[i] is equal to the product of all the elements of nums except nums[i].

The product of any prefix or suffix of nums is guaranteed to fit in a 32-bit integer.

Example 1:

Input: nums = [1,2,3,4]

Output: [24,12,8,6]

Example 2:

Input: nums = [-1,1,0,-3,3]

Output: [0,0,9,0,0]

Constraints:

2 <= nums.length <= 105

-30 <= nums[i] <= 30

The product of any prefix or suffix of nums is guaranteed to fit in a 32-bit integer.

Ques - 2

Given an array nums. We define a running sum of an array as runningSum[i] = sum(nums[0]…nums[i]).

Return the running sum of nums.

Example 1:

Input: nums = [1,2,3,4]

Output: [1,3,6,10]

Explanation: Running sum is obtained as follows: [1, 1+2, 1+2+3, 1+2+3+4].

Example 2:

Input: nums = [1,1,1,1,1]

Output: [1,2,3,4,5]

Explanation: Running sum is obtained as follows: [1, 1+1, 1+1+1, 1+1+1+1, 1+1+1+1+1].

Example 3:

Input: nums = [3,1,2,10,1]

Output: [3,4,6,16,17]

Constraints:

1 <= nums.length <= 1000

-10^6 <= nums[i] <= 10^6

Ques - 3

Given an array nums. Check the given array is in sorted(ascending) format or not . If it is sorted then return true, if it is not then return false.

Input : nums = [1,2,3,4,5]

Output : true

Input : nums = [3,4,4,5,7,12,12,15]

Output : true

Input : nums = [1,2,1,3,4]

Output : false

Ques - 4

Given an arrays nums. Find and return the kth maximum element from an unsorted array(Without sorting), if it is not found return -1.

Hint : Find max and min

Input : nums = [1,3,2,4,5,2,4]

Enter the kth rank : 3

Output : Max is : 3

Input : nums = [10,20,30,10,20,50,30]

Enter the kth rank : 2

Output : Max is : 30

Input : nums = [10,20,10,20,30]

Enter the kth rank : 4

Output : Max is : -1

Day 7(5 Leet Code Questions):

Ques - 1

How Many Numbers Are Smaller Than the Current Number :

Given the array nums, for each nums[i] find out how many numbers in the array are smaller than it. That is, for each nums[i] you have to count the number of valid j's such that j != i and nums[j] < nums[i].

Return the answer in an array.

Example 1:

Input: nums = [8,1,2,2,3]

Output: [4,0,1,1,3]

Explanation:

For nums[0]=8 there exist four smaller numbers than it (1, 2, 2 and 3).

For nums[1]=1 does not exist any smaller number than it.

For nums[2]=2 there exist one smaller number than it (1).

For nums[3]=2 there exist one smaller number than it (1).

For nums[4]=3 there exist three smaller numbers than it (1, 2 and 2).

Example 2:

Input: nums = [6,5,4,8]

Output: [2,1,0,3]

Example 3:

Input: nums = [7,7,7,7]

Output: [0,0,0,0]

Constraints:

2 <= nums.length <= 500

0 <= nums[i] <= 100

Ques - 2

Difference Between Element Sum and Digit Sum of an Array

You are given a positive integer array nums.

The element sum is the sum of all the elements in nums.

The digit sum is the sum of all the digits (not necessarily distinct) that appear in nums.

Return the absolute difference between the element sum and digit sum of nums.

Note that the absolute difference between two integers x and y is defined as |x - y|.

Example 1:

Input: nums = [1,15,6,3]

Output: 9

Explanation:

The element sum of nums is 1 + 15 + 6 + 3 = 25.

The digit sum of nums is 1 + 1 + 5 + 6 + 3 = 16.

The absolute difference between the element sum and digit sum is |25 - 16| = 9.

Example 2:

Input: nums = [1,2,3,4]

Output: 0

Explanation:

The element sum of nums is 1 + 2 + 3 + 4 = 10.

The digit sum of nums is 1 + 2 + 3 + 4 = 10.

The absolute difference between the element sum and digit sum is |10 - 10| = 0.

Constraints:

1 <= nums.length <= 2000

1 <= nums[i] <= 2000

Ques - 3

Sum of All Odd Length Subarrays :

Given an array of positive integers arr, return the sum of all possible odd-length subarrays of arr.

A subarray is a contiguous subsequence of the array.

Example 1:

Input: arr = [1,4,2,5,3]

Output: 58

Explanation: The odd-length subarrays of arr and their sums are:

[1] = 1

[4] = 4

[2] = 2

[5] = 5

[3] = 3

[1,4,2] = 7

[4,2,5] = 11

[2,5,3] = 10

[1,4,2,5,3] = 15

If we add all these together we get 1 + 4 + 2 + 5 + 3 + 7 + 11 + 10 + 15 = 58

Example 2:

Input: arr = [1,2]

Output: 3

Explanation: There are only 2 subarrays of odd length, [1] and [2]. Their sum is 3.

Example 3:

Input: arr = [10,11,12]

Output: 66

Constraints:

1 <= arr.length <= 100

1 <= arr[i] <= 1000

Ques - 4

Find Common Elements Between Two Arrays :

You are given two 0-indexed integer arrays nums1 and nums2 of sizes n and m, respectively.

Consider calculating the following values:

The number of indices i such that 0 <= i < n and nums1[i] occurs at least once in nums2.

The number of indices i such that 0 <= i < m and nums2[i] occurs at least once in nums1.

Return an integer array answer of size 2 containing the two values in the above order.

Example 1:

Input: nums1 = [4,3,2,3,1], nums2 = [2,2,5,2,3,6]

Output: [3,4]

Explanation: We calculate the values as follows:

- The elements at indices 1, 2, and 3 in nums1 occur at least once in nums2. So the first value is 3.

- The elements at indices 0, 1, 3, and 4 in nums2 occur at least once in nums1. So the second value is 4.

Example 2:

Input: nums1 = [3,4,2,3], nums2 = [1,5]

Output: [0,0]

Explanation: There are no common elements between the two arrays, so the two values will be 0.

Constraints:

n == nums1.length

m == nums2.length

1 <= n, m <= 100

1 <= nums1[i], nums2[i] <= 100

Ques - 5

Sum of Squares of Special Elements :

You are given a 1-indexed integer array nums of length n.

An element nums[i] of nums is called special if i divides n, i.e. n % i == 0.

Return the sum of the squares of all special elements of nums.

Example 1:

Input: nums = [1,2,3,4] n = 4

Output: 21

Explanation: There are exactly 3 special elements in nums: nums[1] since 1 divides 4, nums[2] since 2 divides 4, and nums[4] since 4 divides 4.

Hence, the sum of the squares of all special elements of nums is nums[1] * nums[1] + nums[2] * nums[2] + nums[4] * nums[4] = 1 * 1 + 2 * 2 + 4 * 4 = 21.

Example 2:

Input: nums = [2,7,1,19,18,3]   length = 6

Output: 63

Explanation:

There are exactly 4 special elements in nums: nums[1] since 1 divides 6, nums[2] since 2 divides 6, nums[3] since 3 divides 6, and nums[6] since 6 divides 6.

Hence, the sum of the squares of all special elements of nums is nums[1] * nums[1] + nums[2] * nums[2] + nums[3] * nums[3] + nums[6] * nums[6] = 2 * 2 + 7 * 7 + 1 * 1 + 3 * 3 = 63.

Constraints:

1 <= nums.length == n <= 50

1 <= nums[i] <= 50

Day 8(5 Leet Code Questions):

Ques- 1

Maximum Sum With Exactly K Elements

You are given a 0-indexed integer array nums and an integer k. Your task is to perform the following operation exactly k times in order to maximize your score:

Select an element m from nums.

Remove the selected element m from the array.

Add a new element with a value of m + 1 to the array.

Increase your score by m.

Return the maximum score you can achieve after performing the operation exactly k times.

Example 1:

Input: nums = [1,2,3,4,5], k = 3

Output: 18

Explanation: We need to choose exactly 3 elements from nums to maximize the sum.

For the first iteration, we choose 5. Then sum is 5 and nums = [1,2,3,4,6]

For the second iteration, we choose 6. Then sum is 5 + 6 and nums = [1,2,3,4,7]

For the third iteration, we choose 7. Then sum is 5 + 6 + 7 = 18 and nums = [1,2,3,4,8]

So, we will return 18.

It can be proven, that 18 is the maximum answer that we can achieve.

Example 2:

Input: nums = [5,5,5], k = 2

Output: 11

Explanation: We need to choose exactly 2 elements from nums to maximize the sum.

For the first iteration, we choose 5. Then sum is 5 and nums = [5,5,6]

For the second iteration, we choose 6. Then sum is 5 + 6 = 11 and nums = [5,5,7]

So, we will return 11.

It can be proven, that 11 is the maximum answer that we can achieve.

Ques - 2

Check If All 1's Are at Least Length K Places Away

Given an binary array nums and an integer k, return true if all 1's are at least k places away from each other, otherwise return false.

Example 1:

Input: nums = [1,0,0,0,1,0,0,1], k = 2

Output: true

Explanation: Each of the 1s are at least 2 places away from each other.

Example 2:

Input: nums = [1,0,0,1,0,1], k = 2

Output: false

Explanation: The second 1 and third 1 are only one apart from each other.

Ques - 3

The next greater element of some element x in an array is the first greater element that is to the right of x in the same array.

You are given two distinct 0-indexed integer arrays nums1 and nums2, where nums1 is a subset of nums2.

For each 0 <= i < nums1.length, find the index j such that nums1[i] == nums2[j] and determine the next greater element of nums2[j] in nums2. If there is no next greater element, then the answer for this query is -1.

Return an array ans of length nums1.length such that ans[i] is the next greater element as described above.

Example 1:

Input: nums1 = [4,1,2], nums2 = [1,3,4,2]

Output: [-1,3,-1]

Explanation: The next greater element for each value of nums1 is as follows:

- 4 is underlined in nums2 = [1,3,4,2]. There is no next greater element, so the answer is -1.

- 1 is underlined in nums2 = [1,3,4,2]. The next greater element is 3.

- 2 is underlined in nums2 = [1,3,4,2]. There is no next greater element, so the answer is -1.

Example 2:

Input: nums1 = [2,4], nums2 = [1,2,3,4]

Output: [3,-1]

Explanation: The next greater element for each value of nums1 is as follows:

- 2 is underlined in nums2 = [1,2,3,4]. The next greater element is 3.

- 4 is underlined in nums2 = [1,2,3,4]. There is no next greater element, so the answer is -1.

Constraints:

1 <= nums1.length <= nums2.length <= 1000

0 <= nums1[i], nums2[i] <= 104

All integers in nums1 and nums2 are unique.

All the integers of nums1 also appear in nums2

Ques - 4

Number of Students Doing Homework at a Given Time

Given two integer arrays startTime and endTime and given an integer queryTime.

The ith student started doing their homework at the time startTime[i] and finished it at time endTime[i].

Return the number of students doing their homework at time queryTime. More formally, return the number of students where queryTime lays in the interval [startTime[i], endTime[i]] inclusive.

Example 1:

Input: startTime = [1,2,3], endTime = [3,2,7], queryTime = 4

Output: 1

Explanation: We have 3 students where:

The first student started doing homework at time 1 and finished at time 3 and wasn't doing anything at time 4.

The second student started doing homework at time 2 and finished at time 2 and also wasn't doing anything at time 4.

The third student started doing homework at time 3 and finished at time 7 and was the only student doing homework at time 4.

Example 2:

Input: startTime = [4], endTime = [4], queryTime = 4

Output: 1

Explanation: The only student was doing their homework at the queryTime.

Constraints:

startTime.length == endTime.length

1 <= startTime.length <= 100

1 <= startTime[i] <= endTime[i] <= 1000

1 <= queryTime <= 1000

Ques – 5

Final Prices With a Special Discount in a Shop

You are given an integer array prices where prices[i] is the price of the ith item in a shop.

There is a special discount for items in the shop. If you buy the ith item, then you will receive a discount equivalent to prices[j] where j is the minimum index such that j > i and prices[j] <= prices[i]. Otherwise, you will not receive any discount at all.

Return an integer array answer where answer[i] is the final price you will pay for the ith item of the shop, considering the special discount.

Example 1:

Input: prices = [8,4,6,2,3]

Output: [4,2,4,2,3]

Explanation:

For item 0 with price[0]=8 you will receive a discount equivalent to prices[1]=4, therefore, the final price you will pay is 8 - 4 = 4.

For item 1 with price[1]=4 you will receive a discount equivalent to prices[3]=2, therefore, the final price you will pay is 4 - 2 = 2.

For item 2 with price[2]=6 you will receive a discount equivalent to prices[3]=2, therefore, the final price you will pay is 6 - 2 = 4.

For items 3 and 4 you will not receive any discount at all.

Example 2:

Input: prices = [1,2,3,4,5]

Output: [1,2,3,4,5]

Explanation: In this case, for all items, you will not receive any discount at all.

Example 3:

Input: prices = [10,1,1,6]

Output: [9,0,1,6]

Constraints:

1 <= prices.length <= 500

1 <= prices[i] <= 1000

Day 9(4 Leet Code Questions):

Q1)

Intersection of Two Arrays

Given two integer arrays nums1 and nums2, return an array of their

intersection

. Each element in the result must be unique and you may return the result in any order.

Example 1:

Input: nums1 = [1,2,2,1], nums2 = [2,2]

Output: [2]

Example 2:

Input: nums1 = [4,9,5], nums2 = [9,4,9,8,4]

Output: [9,4]

Explanation: [4,9] is also accepted.

Constraints:

1 <= nums1.length, nums2.length <= 1000

0 <= nums1[i], nums2[i] <= 1000

Q2)

Sort Array by Increasing Frequency

Given an array of integers nums, sort the array in increasing order based on the frequency of the values. If multiple values have the same frequency, sort them in decreasing order.

Return the sorted array.

Example 1:

Input: nums = [1,1,2,2,2,3]

Output: [3,1,1,2,2,2]

Explanation: '3' has a frequency of 1, '1' has a frequency of 2, and '2' has a frequency of 3.

Example 2:

Input: nums = [2,3,1,3,2]

Output: [1,3,3,2,2]

Explanation: '2' and '3' both have a frequency of 2, so they are sorted in decreasing order.

Example 3:

Input: nums = [-1,1,-6,4,5,-6,1,4,1]

Output: [5,-1,4,4,-6,-6,1,1,1]

Constraints:

1 <= nums.length <= 100

-100 <= nums[i] <= 100

Q3)

Keep Multiplying Found Values by Two

You are given an array of integers nums. You are also given an integer original which is the first number that needs to be searched for in nums.

You then do the following steps:

If original is found in nums, multiply it by two (i.e., set original = 2 * original).

Otherwise, stop the process.

Repeat this process with the new number as long as you keep finding the number.

Return the final value of original.

Example 1:

Input: nums = [5,3,6,1,12], original = 3

Output: 24

Explanation:

- 3 is found in nums. 3 is multiplied by 2 to obtain 6.

- 6 is found in nums. 6 is multiplied by 2 to obtain 12.

- 12 is found in nums. 12 is multiplied by 2 to obtain 24.

- 24 is not found in nums. Thus, 24 is returned.

Example 2:

Input: nums = [2,7,9], original = 4

Output: 4

Explanation:

- 4 is not found in nums. Thus, 4 is returned.

Q4)

Replace Elements with Greatest Element on Right Side

Given an array arr, replace every element in that array with the greatest element among the elements to its right, and replace the last element with -1.

After doing so, return the array.

Example 1:

Input: arr = [17,18,5,4,6,1]

Output: [18,6,6,6,1,-1]

Explanation:

- index 0 --> the greatest element to the right of index 0 is index 1 (18).

- index 1 --> the greatest element to the right of index 1 is index 4 (6).

- index 2 --> the greatest element to the right of index 2 is index 4 (6).

- index 3 --> the greatest element to the right of index 3 is index 4 (6).

- index 4 --> the greatest element to the right of index 4 is index 5 (1).

- index 5 --> there are no elements to the right of index 5, so we put -1.

Example 2:

Input: arr = [400]

Output: [-1]

Explanation: There are no elements to the right of index 0.

Day 10(5 Leet Code Questions):

Ques - 1

Duplicate Zeros

Given a fixed-length integer array arr, duplicate each occurrence of zero, shifting the remaining elements to the right.

Note that elements beyond the length of the original array are not written. Do the above modifications to the input array in place and do not return anything.

Example 1:

Input: arr = [1,0,2,3,0,4,5,0]

Output: [1,0,0,2,3,0,0,4]

Explanation: After calling your function, the input array is modified to: [1,0,0,2,3,0,0,4]

Example 2:

Input: arr = [1,2,3]

Output: [1,2,3]

Explanation: After calling your function, the input array is modified to: [1,2,3]

Constraints:

1 <= arr.length <= 104

0 <= arr[i] <= 9

Ques - 2

Most Frequent Even Element

Given an integer array nums, return the most frequent even element.

If there is a tie, return the smallest one. If there is no such element, return -1.

Example 1:

Input: nums = [0,1,2,2,4,4,1]

Output: 2

Explanation:

The even elements are 0, 2, and 4. Of these, 2 and 4 appear the most.

We return the smallest one, which is 2.

Example 2:

Input: nums = [4,4,4,9,2,4]

Output: 4

Explanation: 4 is the even element appears the most.

Example 3:

Input: nums = [29,47,21,41,13,37,25,7]

Output: -1

Explanation: There is no even element.

Constraints:

1 <= nums.length <= 2000

0 <= nums[i] <= 105

Ques – 3

Largest Number At Least Twice of Others

You are given an integer array nums where the largest integer is unique.

Determine whether the largest element in the array is at least twice as much as every other number in the array. If it is, return the index of the largest element, or return -1 otherwise.

Example 1:

Input: nums = [3,6,1,0]

Output: 1

Explanation: 6 is the largest integer.

For every other number in the array x, 6 is at least twice as big as x.

The index of value 6 is 1, so we return 1.

Example 2:

Input: nums = [1,2,3,4]

Output: -1

Explanation: 4 is less than twice the value of 3, so we return -1.

Constraints:

2 <= nums.length <= 50

0 <= nums[i] <= 100

The largest element in nums is unique.

Ques - 4

Add to Array-Form of Integer

The array-form of an integer num is an array representing its digits in left to right order.

For example, for num = 1321, the array form is [1,3,2,1].

Given num, the array-form of an integer, and an integer k, return the array-form of the integer num + k.

Example 1:

Input: num = [1,2,0,0], k = 34   1200+34 = 1234

Output: [1,2,3,4]

Explanation: 1200 + 34 = 1234

Example 2:

Input: num = [2,7,4], k = 181

Output: [4,5,5]

Explanation: 274 + 181 = 455

Example 3:

Input: num = [2,1,5], k = 806

Output: [1,0,2,1]

Explanation: 215 + 806 = 1021

Constraints:

1 <= num.length <= 104

0 <= num[i] <= 9

num does not contain any leading zeros except for the zero itself.

1 <= k <= 104

Ques - 5

Plus One

You are given a large integer represented as an integer array digits, where each digits[i] is the ith digit of the integer. The digits are ordered from most significant to least significant in left-to-right order. The large integer does not contain any leading 0's.

Increment the large integer by one and return the resulting array of digits.

Example 1:

Input: digits = [1,2,3]

Output: [1,2,4]

Explanation: The array represents the integer 123.

Incrementing by one gives 123 + 1 = 124.

Thus, the result should be [1,2,4].

Example 2:

Input: digits = [4,3,2,1]

Output: [4,3,2,2]

Explanation: The array represents the integer 4321.

Incrementing by one gives 4321 + 1 = 4322.

Thus, the result should be [4,3,2,2].

Example 3:

Input: digits = [9]

Output: [1,0]

Explanation: The array represents the integer 9.

Incrementing by one gives 9 + 1 = 10.

Thus, the result should be [1,0].

Constraints:

1 <= digits.length <= 100

0 <= digits[i] <= 9

digits does not contain any leading 0's.

Day 11(5 Leet Code Questions):

Ques - 1

Given an array arr[], find the maximum j – i such that arr[i] <= arr[j]

Given an array arr[] of N positive integers. The task is to find the maximum of j – i subjected to the constraint of arr[i] <= arr[j].

Examples :

Input: {34, 8, 10, 3, 2, 80, 30, 33, 1}

Output: 6 (j = 7, i = 1)

Input: {9, 2, 3, 4, 5, 6, 7, 8, 18, 0}

Output: 8 (j = 8, i = 0)

Input: {1, 2, 3, 4, 5, 6}

Output: 5 (j = 5, i = 0)

Input: {6, 5, 4, 3, 2, 1}

Output: 0

Ques - 2

Majority Element

Find the majority element in the array. A majority element in an array A[] of size n is an element that appears more than n/2 times (and hence there is at most one such element).

Examples :

Input : A[]={3, 3, 4, 2, 4, 4, 2, 4, 4}

Output : 4

Explanation: The frequency of 4 is 5 which is greater than the half of the size of the array size.

Input : A[] = {3, 3, 4, 2, 4, 4, 2, 4}

Output : No Majority Element

Explanation: There is no element whose frequency is greater than the half of the size of the array size.

Ques – 3

Sort an array in wave form

Given an unsorted array of integers, sort the array into a wave array. An array arr[0..n-1] is sorted in wave form if:

arr[0] >= arr[1] <= arr[2] >= arr[3] <= arr[4] >= …..

Examples:

Input: arr[] = {10, 5, 6, 3, 2, 20, 100, 80}

Output: arr[] = {3,2,6,5,20,10,100,80}

2,3,5,6,10,20,80,100

Explanation:

here you can see {10, 5, 6, 2, 20, 3, 100, 80} first element is larger than the second and the same thing is repeated again and again. large element – small element-large element -small element and so on .it can be small element-larger element – small element-large element -small element too. all you need to maintain is the up-down fashion which represents a wave. there can be multiple answers.

Input: arr[] = {20, 10, 8, 6, 4, 2}

Output: arr[] = {20, 8, 10, 4, 6, 2}

Ques - 4

Find whether an array is subset of another array

Given two arrays: arr1[0..m-1] and arr2[0..n-1]. Find whether arr2[] is a subset of arr1[] or not. Both arrays are not in sorted order. It may be assumed that elements in both arrays are distinct.

Examples:

Input: arr1[] = {11, 1, 13, 21, 3, 7}, arr2[] = {11, 3, 7, 1}

Output: arr2[] is a subset of arr1[]

Input: arr1[] = {1, 2, 3, 4, 5, 6}, arr2[] = {1, 2, 4}

Output: arr2[] is a subset of arr1[]

Input: arr1[] = {10, 5, 2, 23, 19}, arr2[] = {19, 5, 3}

Output: arr2[] is not a subset of arr1[]

Ques - 5

Subarray with 0 sum

Given an array of positive and negative numbers, the task is to find if there is a subarray (of size at least one) with 0 sum.

Examples:

Input: {4, 2, -3, 1, 6}

Output: true

Explanation:

There is a subarray with zero sum from index 1 to 3.

Input: {4, 2, 0, 1, 6}

Output: true

Explanation: The third element is zero. A single element is also a sub-array.

Input: {-3, 2, 3, 1, 6}

Output: false

Day 12(5 Leet Code Questions):

Ques - 1

Next Greater Element

Given an array arr[ ] of size N having elements, the task is to find the next greater element for each element of the array in order of their appearance in the array.

Next greater element of an element in the array is the nearest element on the right which is greater than the current element.

If there does not exist next greater of current element, then next greater element for current element is -1. For example, next greater of the last element is always -1.

Example 1:

Input:

N = 4, arr[] = [1 3 2 4]

Output:

3 4 4 -1

Explanation:

In the array, the next larger element

to 1 is 3 , 3 is 4 , 2 is 4 and for 4 ?

since it doesn't exist, it is -1.

Example 2:

Input:

N = 5, arr[] [6 8 0 1 3]

Output:

8 -1 1 3 -1

Explanation:

In the array, the next larger element to

6 is 8, for 8 there is no larger elements

hence it is -1, for 0 it is 1 , for 1 it is 3 and then for 3 there is no larger

element on right and hence -1.

Your Task:

This is a function problem. You only need to complete the function nextLargerElement() that takes list of integers arr[ ] and N as input parameters and returns list of integers of length N denoting the next greater elements for all the corresponding elements in the input array.

Expected Time Complexity : O(N)

Expected Auxiliary Space : O(N)

Constraints:

1 ≤ N ≤ 106

0 ≤ Ai ≤ 1018

Ques - 2

Find whether given Array is in form of a mountain or not

Given an array arr[]. The task is to check whether it is a mountain array or not. A mountain array is an array of length at least 3 with elements strictly increasing from starting till an index i, and then strictly decreasing from index i to last index. More formally arr[0] < arr[1] < arr[i] >arr[i+1] > arr[i+2] > arr[N-1].

Examples

Input: arr[] = {4, 4, 3, 2, 1}

Output: false

Input: arr = {1, 2, 3, 4, 9, 8, 7, 6, 5}

Output: true

Ques - 3

Leaders in an array

Given an array A of positive integers. Your task is to find the leaders in the array. An element of array is a leader if it is greater than or equal to all the elements to its right side. The rightmost element is always a leader.

Example 1:

Input:

n = 6

A[] = {16,17,4,3,5,2}

Output: 17 5 2

Explanation: The first leader is 17

as it is greater than all the elements

to its right. Similarly, the next

leader is 5. The right most element

is always a leader so it is also

included.

Example 2:

Input:

n = 5

A[] = {1,2,3,4,0}

Output: 4 0

Explanation: 0 is the rightmost element

and 4 is the only element which is greater

than all the elements to its right.

Your Task:

You don't need to read input or print anything. The task is to complete the function leader() which takes array A and n as input parameters and returns an array of leaders in order of their appearance.

Expected Time Complexity: O(n)

Expected Auxiliary Space: O(n)

Constraints:

1 <= n <= 107

0 <= Ai <= 107

Ques - 4

Find Indexes of a subarray with given sum

Given an unsorted array A of size N that contains only non negative integers, find a continuous sub-array that adds to a given number S and return the left and right index(1-based indexing) of that subarray.

In case of multiple subarrays, return the subarray indexes which come first on moving from left to right.

Note:- You have to return an ArrayList consisting of two elements left and right. In case no such subarray exists return an array consisting of element -1.

Example 1:

Input:

N = 5, S = 12

A[] = {1,2,3,7,5}

Output: 2 4

Explanation: The sum of elements

from 2nd position to 4th position

is 12.

Example 2:

Input:

N = 10, S = 15

A[] = {1,2,3,4,5,6,7,8,9,10}

Output: 1 5

Explanation: The sum of elements

from 1st position to 5th position

is 15.

Your Task:

You don't need to read input or print anything. The task is to complete the function subarraySum() which takes arr, N, and S as input parameters and returns an ArrayList containing the starting and ending positions of the first such occurring subarray from the left where sum equals to S. The two indexes in the array should be according to 1-based indexing. If no such subarray is found, return an array consisting of only one element that is -1.

Expected Time Complexity: O(N)

Expected Auxiliary Space: O(1)

Constraints:

1 <= N <= 105

0 <= Ai <= 109

0<= S <= 109

Ques - 5

Kadane's Algorithm

Given an array Arr[] of N integers. Find the contiguous sub-array(containing at least one number) which has the maximum sum and return its sum.

Example 1:

Input:

N = 5

Arr[] = {1,2,3,-2,5}

Output:

9

Explanation:

Max subarray sum is 9

of elements (1, 2, 3, -2, 5) which

is a contiguous subarray.

Example 2:

Input:

N = 4

Arr[] = {-1,-2,-3,-4}

Output:

-1

Explanation:

Max subarray sum is -1

of element (-1)

Your Task:

You don't need to read input or print anything. The task is to complete the function maxSubarraySum() which takes Arr[] and N as input parameters and returns the sum of subarray with maximum sum.

Expected Time Complexity: O(N)

Expected Auxiliary Space: O(1)

Constraints:

1 ≤ N ≤ 106

-107 ≤ A[i] ≤ 107

Strings(109 Questions)

Day 1(Basic 10 questions):

1.Write a program to print a string entered by user.

2.Write a program to input and display the sentence I love candies.

3.Write a program to find the length of the string "refrigerator".

4.Write a program to check if the letter 'e' is present in the word 'Umbrella'.

5.Write a program to check if the word 'orange' is present in the "This is orange juice".

6.Write a program to find the first and the last occurence of the letter 'o' and character ',' in "Hello, World".

7.Write a program that takes your full name as input and displays the abbreviations of the first and middle names except the last name which is displayed as it is. For example, if your name is Robert Brett Roser, then the output should be R.B.Roser.

8.Write a program to find the number of vowels, consonents, digits and white space characters in a string.

9.Write a program to delete all consonents from the string "Hello, have a good day".

10.Input a string of alphabets. Find out the number of occurrence of all alphabets in that string. Find out the alphabet with maximum occurrence.

Day 2(Basic 10 questions):

Q1) Check the String is empty or not?

Q2) Finding length of String?

Q3) Printing String object?

Q4) Comparing String objects?

Q5) Reading character at the given index

Q6) Finding index of given character

Q7) Check the String starts with given character.

Q8) Check the given String ends with given character.

Q9) Concatenating two different String.

Q10) Replacing old character with new character.

Day 3(Basic 30 questions):

Some Programming Questions on String Predefined methods:

1.Write a Java program to check if a given string is empty or not using the isEmpty() method.

2.Write a Java program to concatenate two strings using the concat() method.

3.Write a Java program to find the length of a string using the length() method.

4.Write a Java program to convert a string to lowercase using the toLowerCase() method.

5.Write a Java program to convert a string to uppercase using the toUpperCase() method.

6.Write a Java program to check if a string contains another string using the contains() method.

7.Write a Java program to extract a portion of a string using the substring() method.

8.Write a Java program to replace all occurrences of a specified value in a string using the replace() method.

9.Write a Java program to split a string into an array of substrings using the split() method.

10.Write a Java program to find the index of the first occurrence of a character in a string using the indexOf() method.

11.Write a Java program to compare two strings lexicographically using the compareTo() method.

12.Write a Java program to remove leading and trailing whitespace from a string using the trim() method.

13.Write a Java program to check if two strings are equal ignoring their case using the equalsIgnoreCase() method.

14.Write a Java program to check if a string starts with a specific prefix using the startsWith() method.

15.Write a Java program to check if a string ends with a specific suffix using the endsWith() method.

16.Write a Java program to find the index of the last occurrence of a character in a string using the lastIndexOf() method.

17.Write a Java program to count the occurrences of a specific substring in a string using the indexOf() method in a loop.

18.Write a Java program to reverse a string using the StringBuilder class or by implementing your own logic.

19.Write a Java program to split a string into an array of characters using the toCharArray() method.

20.Write a Java program to convert an integer to a string using the valueOf() method of the String class.

21.Write a Java program to check if a string contains only numeric characters using the matches() method with a regular expression.

22.Write a Java program to find the longest word in a given string using the split() method and iteration.

23.Write a Java program to count the number of vowels in a string using the charAt() method and a loop.

24.Write a Java program to remove all occurrences of a specified character from a string using the replace() method in a loop.

25.Write a Java program to check if a string is a palindrome (reads the same backward as forward) using string manipulation methods.

26.Write a Java program to capitalize the first letter of each word in a sentence using the split() method and string manipulation.

28.Write a Java program to check if a string is an anagram of another string using the toCharArray() method and sorting.

29.Write a Java program to remove duplicate characters from a string using the HashSet or an array.

30.Write a Java program to reverse the order of words in a sentence using the split() method and iteration.

Day 4(Basic 10 questions):

Ques-1

Finding First Unique Character:

Given a String, write a function that finds the first character that appears only once in the String. If all characters are repeated or the String is empty, return a special character (e.g., '-').

Example: "leetcode" returns 'l', but "aabbcc" returns '-'.

Ques-2

Given a String, write a method that compresses the String by consecutively repeating characters. If a character only appears once, it shouldn't be compressed.

Example: "aabcccccaaa" compresses to "a2c5a3"

Ques-3

Counting Vowels:

Given a String, write a function that counts the number of vowels (a, e, i, o, u) present in the String (case-insensitive).

Example: "Hello world!" has 3 vowels.

Ques-4

Removing Duplicates:

Given a String, write a method that removes all duplicate characters from the String while maintaining the relative order of the remaining characters.

Example: "abbcccddd" becomes "abcd".

Ques-5

Finding Most Frequent Character:

Given a String, write a function that finds the character that appears the most frequently in the String.

Example: "aaabbcc" has 'a' as the most frequent character.

Ques-6

First and Last Character Swap:

Given a String, write a function that swaps the first and last characters and returns the new String.

Example: "apple" becomes "leapp".

Ques-7

Remove All Vowels:

Given a String, write a function that removes all vowels (a, e, i, o, u) and returns the modified String.

Example: "hello world!" becomes "hll wrld!"

Ques-8

Alternate Character Removal:

Given a String, write a function that removes every other character, starting from the second character, and returns the modified String.

Example: "Hello world!" becomes "Hlo ol!" (keeps first character, removes second, keeps third, and so on).

Ques-9

Find the Middle Character:

Given a String, write a function that finds and returns the middle character. If the String has an even number of characters, return the concatenation of the two middle characters.

Example: "hello" returns 'l', "worlld" returns "rl" (concatenation of middle two).

Ques-10

Count Words Starting with a Vowel:

Given a String with words separated by spaces, write a function that counts the number of words that start with a vowel (a, e, i, o, u), ignoring case.

Example: apple is good for eyes.

Output: 3

Day 5(Basic Leet code 5 questions):

Ques - 1

Find First Palindromic String in the Array

Given an array of strings words, return the first palindromic string in the array. If there is no such string, return an empty string "".

A string is palindromic if it reads the same forward and backward.

Example 1:

Input: words = ["abc","car","ada","racecar","cool"]

Output: "ada"

Explanation: The first string that is palindromic is "ada".

Note that "racecar" is also palindromic, but it is not the first.

Example 2:

Input: words = ["notapalindrome","racecar"]

Output: "racecar"

Explanation: The first and only string that is palindromic is "racecar".

Example 3:

Input: words = ["def","ghi"]

Output: ""

Explanation: There are no palindromic strings, so the empty string is returned.

Constraints:

1 <= words.length <= 100

1 <= words[i].length <= 100

words[i] consists only of lowercase English letters

Ques- 2

Faulty Keyboard

Your laptop keyboard is faulty, and whenever you type a character 'i' on it, it reverses the string that you have written. Typing other characters works as expected.

You are given a 0-indexed string s, and you type each character of s using your faulty keyboard.

Return the final string that will be present on your laptop screen.

Example 1:

Input: s = "string"

Output: "rtsng"

Explanation:

After typing first character, the text on the screen is "s".

After the second character, the text is "st".

After the third character, the text is "str".

Since the fourth character is an 'i', the text gets reversed and becomes "rts".

After the fifth character, the text is "rtsn".

After the sixth character, the text is "rtsng".

Therefore, we return "rtsng".

Example 2:

Input: s = "poiinter"

Output: "ponter"

Explanation:

After the first character, the text on the screen is "p".

After the second character, the text is "po".

Since the third character you type is an 'i', the text gets reversed and becomes "op".

Since the fourth character you type is an 'i', the text gets reversed and becomes "po".

After the fifth character, the text is "pon".

After the sixth character, the text is "pont".

After the seventh character, the text is "ponte".

After the eighth character, the text is "ponter".

Therefore, we return "ponter".

Constraints:

1 <= s.length <= 100

s consists of lowercase English letters.

s[0] != 'i'

Ques - 3

To Lower Case

Given a string s, return the string after replacing every uppercase letter with the same lowercase letter.

Example 1:

Input: s = "Hello"

Output: "hello"

Example 2:

Input: s = "here"

Output: "here"

Example 3:

Input: s = "LOVELY"

Output: "lovely"

Constraints:

1 <= s.length <= 100

s consists of printable ASCII characters.

Ques - 4

Check if the Sentence Is Pangram

A pangram is a sentence where every letter of the English alphabet appears at least once.

Given a string sentence containing only lowercase English letters, return true if sentence is a pangram, or false otherwise.

Example 1:

Input: sentence = "thequickbrownfoxjumpsoverthelazydog"

Output: true

Explanation: sentence contains at least one of every letter of the English alphabet.

Example 2:

Input: sentence = "leetcode"

Output: false

Constraints:

1 <= sentence.length <= 1000

sentence consists of lowercase English letters

Ques – 5

Reverse Words in a String III

Given a string s, reverse the order of characters in each word within a sentence while still preserving whitespace and initial word order.

Example 1:

Input: s = "Let's take LeetCode contest"

Output: "s'teL ekat edoCteeL tsetnoc"

Example 2:

Input: s = "Mr Ding"

Output: "rM gniD"

Constraints:

1 <= s.length <= 5 * 104

s contains printable ASCII characters.

s does not contain any leading or trailing spaces.

There is at least one word in s.

All the words in s are separated by a single space.

Day 6(Basic Leet code 4 questions):

Ques - 1

Sorting the Sentence

A sentence is a list of words that are separated by a single space with no leading or trailing spaces. Each word consists of lowercase and uppercase English letters.

A sentence can be shuffled by appending the 1-indexed word position to each word then rearranging the words in the sentence.

For example, the sentence "This is a sentence" can be shuffled as "sentence4 a3 is2 This1" or "is2 sentence4 This1 a3".

Given a shuffled sentence s containing no more than 9 words, reconstruct and return the original sentence.

Example 1:

Input: s = "is2 sentence4 This1 a3"

Output: "This is a sentence"

Explanation: Sort the words in s to their original positions "This1 is2 a3 sentence4", then remove the numbers.

Example 2:

Input: s = "Myself2 Me1 I4 and3"

Output: "Me Myself and I"

Explanation: Sort the words in s to their original positions "Me1 Myself2 and3 I4", then remove the numbers.

Constraints:

2 <= s.length <= 200

s consists of lowercase and uppercase English letters, spaces, and digits from 1 to 9.

The number of words in s is between 1 and 9.

The words in s are separated by a single space.

s contains no leading or trailing spaces

Ques - 2

Reverse Words in a String III

Given a string s, reverse the order of characters in each word within a sentence while still preserving whitespace and initial word order.

Example 1:

Input: s = "Let's take LeetCode contest"

Output: "s'teL ekat edoCteeL tsetnoc"

Example 2:

Input: s = "Mr Ding"

Output: "rM gniD"

Constraints:

1 <= s.length <= 5 * 104

s contains printable ASCII characters.

s does not contain any leading or trailing spaces.

There is at least one word in s.

All the words in s are separated by a single space.

Ques - 3

Count the Number of Consistent Strings

You are given a string allowed consisting of distinct characters and an array of strings words. A string is consistent if all characters in the string appear in the string allowed.

Return the number of consistent strings in the array words.

Example 1:

Input: allowed = "ab", words = ["ad","bd","aaab","baa","badab"]

Output: 2

Explanation: Strings "aaab" and "baa" are consistent since they only contain characters 'a' and 'b'.

Example 2:

Input: allowed = "abc", words = ["a","b","c","ab","ac","bc","abc"]

Output: 7

Explanation: All strings are consistent.

Example 3:

Input: allowed = "cad", words = ["cc","acd","b","ba","bac","bad","ac","d"]

Output: 4

Explanation: Strings "cc", "acd", "ac", and "d" are consistent.

Constraints:

1 <= words.length <= 104

1 <= allowed.length <= 26

1 <= words[i].length <= 10

The characters in allowed are distinct.

words[i] and allowed contain only lowercase English letters

Ques - 4

Check if a String Is an Acronym of Words

Given an array of strings words and a string s, determine if s is an acronym of words.

The string s is considered an acronym of words if it can be formed by concatenating the first character of each string in words in order. For example, "ab" can be formed from ["apple", "banana"], but it can't be formed from ["bear", "aardvark"].

Return true if s is an acronym of words, and false otherwise.

Example 1:

Input: words = ["alice","bob","charlie"], s = "abc"

Output: true

Explanation: The first character in the words "alice", "bob", and "charlie" are 'a', 'b', and 'c', respectively. Hence, s = "abc" is the acronym.

Example 2:

Input: words = ["an","apple"], s = "a"

Output: false

Explanation: The first character in the words "an" and "apple" are 'a' and 'a', respectively.

The acronym formed by concatenating these characters is "aa".

Hence, s = "a" is not the acronym.

Example 3:

Input: words = ["never","gonna","give","up","on","you"], s = "ngguoy"

Output: true

Explanation: By concatenating the first character of the words in the array, we get the string "ngguoy".

Hence, s = "ngguoy" is the acronym

Day 7(Leet code 5 questions):

Ques-1

Lexicographically Smallest Palindrome

You are given a string s consisting of lowercase English letters, and you are allowed to perform operations on it. In one operation, you can replace a character in s with another lowercase English letter.

Your task is to make s a palindrome with the minimum number of operations possible. If there are multiple palindromes that can be made using the minimum number of operations, make the lexicographically smallest one.

A string a is lexicographically smaller than a string b (of the same length) if in the first position where a and b differ, string a has a letter that appears earlier in the alphabet than the corresponding letter in b.

Return the resulting palindrome string.

Example 1:

Input: s = "egcfe"

Output: "efcfe"

Explanation: The minimum number of operations to make "egcfe" a palindrome is 1, and the lexicographically smallest palindrome string we can get by modifying one character is "efcfe", by changing 'g'.

Example 2:

Input: s = "abcd"

Output: "abba"

Explanation: The minimum number of operations to make "abcd" a palindrome is 2, and the lexicographically smallest palindrome string we can get by modifying two characters is "abba".

Example 3:

Input: s = "seven"

Output: "neven"

Explanation: The minimum number of operations to make "seven" a palindrome is 1, and the lexicographically smallest palindrome string we can get by modifying one character is "neven".

Constraints:

1 <= s.length <= 1000

s consists of only lowercase English letters

Ques - 2

Replace All Digits with Characters

You are given a 0-indexed string s that has lowercase English letters in its even indices and digits in its odd indices.

There is a function shift(c, x), where c is a character and x is a digit, that returns the xth character after c.

For example, shift('a', 5) = 'f' and shift('x', 0) = 'x'.

For every odd index i, you want to replace the digit s[i] with shift(s[i-1], s[i]).

Return s after replacing all digits. It is guaranteed that shift(s[i-1], s[i]) will never exceed 'z'.

Example 1:

Input: s = "a1c1e1"

Output: "abcdef"

Explanation: The digits are replaced as follows:

- s[1] -> shift('a',1) = 'b'

- s[3] -> shift('c',1) = 'd'

- s[5] -> shift('e',1) = 'f'

Example 2:

Input: s = "a1b2c3d4e"

Output: "abbdcfdhe"

Explanation: The digits are replaced as follows:

- s[1] -> shift('a',1) = 'b'
- s[3] -> shift('b',2) = 'd'
- s[5] -> shift('c',3) = 'f'
- s[7] -> shift('d',4) = 'h'

Constraints:

1 <= s.length <= 100

s consists only of lowercase English letters and digits.

shift(s[i-1], s[i]) <= 'z' for all odd indices i

Ques - 3

Number of Changing Keys

You are given a 0-indexed string s typed by a user. Changing a key is defined as using a key different from the last used key. For example, s = "ab" has a change of a key while s = "bBBb" does not have any.

Return the number of times the user had to change the key.

Note: Modifiers like shift or caps lock won't be counted in changing the key that is if a user typed the letter 'a' and then the letter 'A' then it will not be considered as a changing of key.

Example 1:

Input: s = "aAbBcC"

Output: 2

Explanation:

From s[0] = 'a' to s[1] = 'A', there is no change of key as caps lock or shift is not counted.

From s[1] = 'A' to s[2] = 'b', there is a change of key.

From s[2] = 'b' to s[3] = 'B', there is no change of key as caps lock or shift is not counted.

From s[3] = 'B' to s[4] = 'c', there is a change of key.

From s[4] = 'c' to s[5] = 'C', there is no change of key as caps lock or shift is not counted.

Example 2:

Input: s = "AaAaAaaA"

Output: 0

Explanation: There is no change of key since only the letters 'a' and 'A' are pressed which does not require change of key.

Ques - 4

Sort the People

You are given an array of strings names, and an array heights that consists of distinct positive integers. Both arrays are of length n.

For each index i, names[i] and heights[i] denote the name and height of the ith person.

Return names sorted in descending order by the people's heights.

Example 1:

Input: names = ["Mary","John","Emma"], heights = [180,165,170]

Output: ["Mary","Emma","John"]

Explanation: Mary is the tallest, followed by Emma and John.

Example 2:

Input: names = ["Alice","Bob","Bob"], heights = [155,185,150]

Output: ["Bob","Alice","Bob"]

Explanation: The first Bob is the tallest, followed by Alice and the second Bob.

Constraints:

n == names.length == heights.length

1 <= n <= 103

1 <= names[i].length <= 20

1 <= heights[i] <= 105

names[i] consists of lower and upper case English letters.

All the values of heights are distinct

Ques - 5

Merge Strings Alternately

You are given two strings word1 and word2. Merge the strings by adding letters in alternating order, starting with word1. If a string is longer than the other, append the additional letters onto the end of the merged string.

Return the merged string.

Example 1:

Input: word1 = "abc", word2 = "pqr"

Output: "apbqcr"

Explanation: The merged string will be merged as so:

```
word1:  a   b   c
word2:    p   q   r
merged: a p b q c r
```

Example 2:

Input: word1 = "ab", word2 = "pqrs"

Output: "apbqrs"

Explanation: Notice that as word2 is longer, "rs" is appended to the end.

```
word1:  a   b
word2:    p   q   r   s
merged: a p b q   r   s
```

Example 3:

Input: word1 = "abcd", word2 = "pq"

Output: "apbqcd"

Explanation: Notice that as word1 is longer, "cd" is appended to the end.

```
word1:  a   b   c   d
word2:    p   q
merged: a p b q c   d
```

Constraints:

1 <= word1.length, word2.length <= 100

word1 and word2 consist of lowercase English letters

Day 8(Leet code 5 questions):

Ques - 1

Score of a String

You are given a string s. The score of a string is defined as the sum of the absolute difference between the ASCII values of adjacent characters.

Return the score of s.

Example 1:

Input: s = "hello"

Output: 13

Explanation:

The ASCII values of the characters in s are: 'h' = 104, 'e' = 101, 'l' = 108, 'o' = 111. So, the score of s would be |104 - 101| + |101 - 108| + |108 - 108| + |108 - 111| = 3 + 7 + 0 + 3 = 13.

Example 2:

Input: s = "zaz"

Output: 50

Explanation:

The ASCII values of the characters in s are: 'z' = 122, 'a' = 97. So, the score of s would be |122 - 97| + |97 - 122| = 25 + 25 = 50.

Constraints:

2 <= s.length <= 100

s consists only of lowercase English letters.

Ques - 2

Defanging an IP Address

Given a valid (IPv4) IP address, return a defanged version of that IP address.

A defanged IP address replaces every period "." with "[.]".

Example 1:

Input: address = "1.1.1.1"

Output: "1[.]1[.]1[.]1"

Example 2:

Input: address = "255.100.50.0"

Output: "255[.]100[.]50[.]0"

Constraints:

The given address is a valid IPv4 address.

Ques - 3

Final Value of Variable After Performing Operations

There is a programming language with only four operations and one variable X:

++X and X++ increments the value of the variable X by 1.

--X and X-- decrements the value of the variable X by 1.

Initially, the value of X is 0.

Given an array of strings operations containing a list of operations, return the final value of X after performing all the operations.

Example 1:

Input: operations = ["--X","X++","X++"]

Output: 1

Explanation: The operations are performed as follows:

Initially, X = 0.

--X: X is decremented by 1, X = 0 - 1 = -1.

X++: X is incremented by 1, X = -1 + 1 = 0.

X++: X is incremented by 1, X = 0 + 1 = 1.

Example 2:

Input: operations = ["++X","++X","X++"]

Output: 3

Explanation: The operations are performed as follows:

Initially, X = 0.

++X: X is incremented by 1, X = 0 + 1 = 1.

++X: X is incremented by 1, X = 1 + 1 = 2.

X++: X is incremented by 1, X = 2 + 1 = 3.

Example 3:

Input: operations = ["X++","++X","--X","X--"]

Output: 0

Explanation: The operations are performed as follows:

Initially, X = 0.

X++: X is incremented by 1, X = 0 + 1 = 1.

++X: X is incremented by 1, X = 1 + 1 = 2.

--X: X is decremented by 1, X = 2 - 1 = 1.

X--: X is decremented by 1, X = 1 - 1 = 0.

Constraints:

1 <= operations.length <= 100

operations[i] will be either "++X", "X++", "--X", or "X--"

Ques - 4

Jewels and Stones

You're given strings jewels representing the types of stones that are jewels, and stones representing the stones you have. Each character in stones is a type of stone you have. You want to know how many of the stones you have are also jewels.

Letters are case sensitive, so "a" is considered a different type of stone from "A".

Example 1:

Input: jewels = "aA", stones = "aAAbbbb"

Output: 3

Example 2:

Input: jewels = "z", stones = "ZZ"

Output: 0

Constraints:

1 <= jewels.length, stones.length <= 50

jewels and stones consist of only English letters.

All the characters of jewels are unique.

Ques - 5

Maximum Number of Words Found in Sentences

A sentence is a list of words that are separated by a single space with no leading or trailing spaces.

You are given an array of strings sentences, where each sentences[i] represents a single sentence.

Return the maximum number of words that appear in a single sentence.

Example 1:

Input: sentences = ["alice and bob love leetcode", "i think so too", "this is great thanks very much"]

Output: 6

Explanation:

- The first sentence, "alice and bob love leetcode", has 5 words in total.

- The second sentence, "i think so too", has 4 words in total.

- The third sentence, "this is great thanks very much", has 6 words in total.

Thus, the maximum number of words in a single sentence comes from the third sentence, which has 6 words.

Example 2:

Input: sentences = ["please wait", "continue to fight", "continue to win"]

Output: 3

Explanation: It is possible that multiple sentences contain the same number of words.

In this example, the second and third sentences (underlined) have the same number of words.

Constraints:

1 <= sentences.length <= 100

1 <= sentences[i].length <= 100

sentences[i] consists only of lowercase English letters and ' ' only.

sentences[i] does not have leading or trailing spaces.

All the words in sentences[i] are separated by a single space.

Day 9(Leet code 5 questions):

Ques - 1

Reverse Prefix of Word

Given a 0-indexed string word and a character ch, reverse the segment of word that starts at index 0 and ends at the index of the first occurrence of ch (inclusive). If the character ch does not exist in word, do nothing.

For example, if word = "abcdefd" and ch = "d", then you should reverse the segment that starts at 0 and ends at 3 (inclusive). The resulting string will be "dcbaefd".

Return the resulting string.

Example 1:

Input: word = "abcdefd", ch = "d"

Output: "dcbaefd"

Explanation: The first occurrence of "d" is at index 3.

Reverse the part of word from 0 to 3 (inclusive), the resulting string is "dcbaefd".

Example 2:

Input: word = "xyxzxe", ch = "z"

Output: "zxyxxe"

Explanation: The first and only occurrence of "z" is at index 3.

Reverse the part of word from 0 to 3 (inclusive), the resulting string is "zxyxxe".

Example 3:

Input: word = "abcd", ch = "z"

Output: "abcd"

Explanation: "z" does not exist in word.

You should not do any reverse operation, the resulting string is "abcd".

Constraints:

1 <= word.length <= 250

word consists of lowercase English letters.

ch is a lowercase English letter.

Ques - 2

Split a String in Balanced Strings

Balanced strings are those that have an equal quantity of 'L' and 'R' characters.

Given a balanced string s, split it into some number of substrings such that:

Each substring is balanced.

Return the maximum number of balanced strings you can obtain.

Example 1:

Input: s = "RLRRLLRLRL"

Output: 4

Explanation: s can be split into "RL", "RRLL", "RL", "RL", each substring contains same number of 'L' and 'R'.

Example 2:

Input: s = "RLRRRLLRLL"

Output: 2

Explanation: s can be split into "RL", "RRRLLRLL", each substring contains same number of 'L' and 'R'.

Note that s cannot be split into "RL", "RR", "RL", "LR", "LL", because the 2nd and 5th substrings are not balanced.

Example 3:

Input: s = "LLLLRRRR"

Output: 1

Explanation: s can be split into "LLLLRRRR".

Constraints:

2 <= s.length <= 1000

s[i] is either 'L' or 'R'.

s is a balanced string

Ques - 3

Check If Two String Arrays are Equivalent

Given two string arrays word1 and word2, return true if the two arrays represent the same string, and false otherwise.

A string is represented by an array if the array elements concatenated in order forms the string.

Example 1:

Input: word1 = ["ab", "c"], word2 = ["a", "bc"]

Output: true

Explanation:

word1 represents string "ab" + "c" -> "abc"

word2 represents string "a" + "bc" -> "abc"

The strings are the same, so return true.

Example 2:

Input: word1 = ["a", "cb"], word2 = ["ab", "c"]

Output: false

Example 3:

Input: word1 = ["abc", "d", "defg"], word2 = ["abcddefg"]

Output: true

Constraints:

1 <= word1.length, word2.length <= 103

1 <= word1[i].length, word2[i].length <= 103

1 <= sum(word1[i].length), sum(word2[i].length) <= 103

word1[i] and word2[i] consist of lowercase letters

Ques - 4

Truncate Sentence

A sentence is a list of words that are separated by a single space with no leading or trailing spaces. Each of the words consists of only uppercase and lowercase English letters (no punctuation).

For example, "Hello World", "HELLO", and "hello world hello world" are all sentences.

You are given a sentence s and an integer k. You want to truncate s such that it contains only the first k words. Return s after truncating it.

Example 1:

Input: s = "Hello how are you Contestant", k = 4

Output: "Hello how are you"

Explanation:

The words in s are ["Hello", "how" "are", "you", "Contestant"].

The first 4 words are ["Hello", "how", "are", "you"].

Hence, you should return "Hello how are you".

Example 2:

Input: s = "What is the solution to this problem", k = 4

Output: "What is the solution"

Explanation:

The words in s are ["What", "is" "the", "solution", "to", "this", "problem"].

The first 4 words are ["What", "is", "the", "solution"].

Hence, you should return "What is the solution".

Example 3:

Input: s = "chopper is not a tanuki", k = 5

Output: "chopper is not a tanuki"

Constraints:

1 <= s.length <= 500

k is in the range [1, the number of words in s].

s consist of only lowercase and uppercase English letters and spaces.

The words in s are separated by a single space.

There are no leading or trailing spaces

Ques - 5

Shuffle String

You are given a string s and an integer array indices of the same length. The string s will be shuffled such that the character at the ith position moves to indices[i] in the shuffled string.

Return the shuffled string.

4 5 6 7 0 2 1 3

c o d e l e e t

After shuffled :

0 1 2 3 4 5 6 7

l e e t c o d e

Example 1:

Input: s = "codeleet", indices = [4,5,6,7,0,2,1,3]

Output: "leetcode"

Explanation: As shown, "codeleet" becomes "leetcode" after shuffling.

Example 2:

Input: s = "abc", indices = [0,1,2]

Output: "abc"

Explanation: After shuffling, each character remains in its position.

Constraints:

s.length == indices.length == n

1 <= n <= 100

s consists of only lowercase English letters.

0 <= indices[i] < n

All values of indices are unique

Day 10(Leet code 5 questions):

Ques - 1

Determine if String Halves Are Alike.

You are given a string s of even length. Split this string into two halves of equal lengths, and let a be the first half and b be the second half.

Two strings are alike if they have the same number of vowels ('a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U'). Notice that s contains uppercase and lowercase letters.

Return true if a and b are alike. Otherwise, return false.

Example 1:

Input: s = "book"

Output: true

Explanation: a = "bo" and b = "ok". a has 1 vowel and b has 1 vowel. Therefore, they are alike.

Example 2:

Input: s = "textbook"

Output: false

Explanation: a = "text" and b = "book". a has 1 vowel whereas b has 2. Therefore, they are not alike.

Notice that the vowel o is counted twice.

Constraints:

2 <= s.length <= 1000

s.length is even.

s consists of uppercase and lowercase letters.

Ques - 2

Remove Trailing Zeros From a String

Given a positive integer num represented as a string, return the integer num without trailing zeros as a string.

Example 1:

Input: num = "51230100"

Output: "512301"

Explanation: Integer "51230100" has 2 trailing zeros, we remove them and return integer "512301".

Example 2:

Input: num = "123"

Output: "123"

Explanation: Integer "123" has no trailing zeros, we return integer "123".

Constraints:

1 <= num.length <= 1000

num consists of only digits.

num doesn't have any leading zeros.

Ques - 3

Counting Words With a Given Prefix

You are given an array of strings words and a string pref.

Return the number of strings in words that contain pref as a prefix.

A prefix of a string s is any leading contiguous substring of s.

Example 1:

Input: words = ["pay","attention","practice","attend"], pref = "at"

Output: 2

Explanation: The 2 strings that contain "at" as a prefix are: "attention" and "attend".

Example 2:

Input: words = ["leetcode","win","loops","success"], pref = "code"

Output: 0

Explanation: There are no strings that contain "code" as a prefix.

Constraints:

1 <= words.length <= 100

1 <= words[i].length, pref.length <= 100

words[i] and pref consist of lowercase English letters.

Ques - 4

Generate a String With Characters That Have Odd Counts

Given an integer n, return a string with n characters such that each character in such string occurs an odd number of times.

The returned string must contain only lowercase English letters. If there are multiples valid strings, return any of them.

Example 1:

Input: n = 4

Output: "pppz"

Explanation: "pppz" is a valid string since the character 'p' occurs three times and the character 'z' occurs once. Note that there are many other valid strings such as "ohhh" and "love".

Example 2:

Input: n = 2

Output: "xy"

Explanation: "xy" is a valid string since the characters 'x' and 'y' occur once. Note that there are many other valid strings such as "ag" and "ur".

Example 3:

Input: n = 7

Output: "holasss"

Constraints:

1 <= n <= 500

Ques - 5

Check if All Characters Have Equal Number of Occurrences

Given a string s, return true if s is a good string, or false otherwise.

A string s is good if all the characters that appear in s have the same number of occurrences (i.e., the same frequency).

Example 1:

Input: s = "abacbc"

Output: true

Explanation: The characters that appear in s are 'a', 'b', and 'c'. All characters occur 2 times in s.

Example 2:

Input: s = "aaabb"

Output: false

Explanation: The characters that appear in s are 'a' and 'b'.

'a' occurs 3 times while 'b' occurs 2 times, which is not the same number of times.

Constraints:

1 <= s.length <= 1000

s consists of lowercase English letters.

Day 11(Leet code 5 questions):

Ques - 1

Increasing Decreasing String

You are given a string s. Reorder the string using the following algorithm:

Pick the smallest character from s and append it to the result.

Pick the smallest character from s which is greater than the last appended character to the result and append it.

Repeat step 2 until you cannot pick more characters.

Pick the largest character from s and append it to the result.

Pick the largest character from s which is smaller than the last appended character to the result and append it.

Repeat step 5 until you cannot pick more characters.

Repeat the steps from 1 to 6 until you pick all characters from s.

In each step, If the smallest or the largest character appears more than once you can choose any occurrence and append it to the result.

Return the result string after sorting s with this algorithm.

Example 1:

Input: s = "aaaabbbbcccc"

Output: "abccbaabccba"

Explanation: After steps 1, 2 and 3 of the first iteration, result = "abc"

After steps 4, 5 and 6 of the first iteration, result = "abccba"

First iteration is done. Now s = "aabbcc" and we go back to step 1

After steps 1, 2 and 3 of the second iteration, result = "abccbaabc"

After steps 4, 5 and 6 of the second iteration, result = "abccbaabccba"

Example 2:

Input: s = "rat"

Output: "art"

Explanation: The word "rat" becomes "art" after re-ordering it with the mentioned algorithm.

Constraints:

1 <= s.length <= 500

s consists of only lowercase English letters

Ques - 2

Minimize String Length

Given a 0-indexed string s, repeatedly perform the following operation any number of times:

Choose an index i in the string, and let c be the character in position i. Delete the closest occurrence of c to the left of i (if any) and the closest occurrence of c to the right of i (if any).

Your task is to minimize the length of s by performing the above operation any number of times.

Return an integer denoting the length of the minimized string.

Example 1:

Input: s = "aaabc"

Output: 3

Explanation: In this example, s is "aaabc". We can start by selecting the character 'a' at index 1. We then remove the closest 'a' to the left of index 1, which is at index 0, and the closest 'a' to the right of index 1, which is at index 2. After this operation, the string becomes "abc". Any further operation we perform on the string will leave it unchanged. Therefore, the length of the minimized string is 3.

Example 2:

Input: s = "cbbd"

Output: 3

Explanation: For this we can start with character 'b' at index 1. There is no occurrence of 'b' to the left of index 1, but there is one to the right at index 2, so we delete the 'b' at index 2. The string becomes "cbd" and further operations will leave it unchanged. Hence, the minimized length is 3.

Example 3:

Input: s = "dddaaa"

Output: 2

Explanation: For this, we can start with the character 'd' at index 1. The closest occurrence of a 'd' to its left is at index 0, and the closest occurrence of a 'd' to its right is at index 2. We delete both index 0 and 2, so the string becomes "daaa". In the new string, we can select the character 'a' at index 2. The closest occurrence of an 'a' to its left is at index 1, and the closest occurrence of an 'a' to its right is at index 3. We delete both of them, and the string becomes "da". We cannot minimize this further, so the minimized length is 2.

Constraints:

1 <= s.length <= 100

s contains only lowercase English letters

Ques - 3

Check if Word Equals Summation of Two Words

The letter value of a letter is its position in the alphabet starting from 0 (i.e. 'a' -> 0, 'b' -> 1, 'c' -> 2, etc.).

The numerical value of some string of lowercase English letters s is the concatenation of the letter values of each letter in s, which is then converted into an integer.

For example, if s = "acb", we concatenate each letter's letter value, resulting in "021". After converting it, we get 21.

You are given three strings firstWord, secondWord, and targetWord, each consisting of lowercase English letters 'a' through 'j' inclusive.

Return true if the summation of the numerical values of firstWord and secondWord equals the numerical value of targetWord, or false otherwise.

Example 1:

Input: firstWord = "acb", secondWord = "cba", targetWord = "cdb"

Output: true

Explanation:

The numerical value of firstWord is "acb" -> "021" -> 21.

The numerical value of secondWord is "cba" -> "210" -> 210.

The numerical value of targetWord is "cdb" -> "231" -> 231.

We return true because 21 + 210 == 231.

Example 2:

Input: firstWord = "aaa", secondWord = "a", targetWord = "aab"

Output: false

Explanation:

The numerical value of firstWord is "aaa" -> "000" -> 0.

The numerical value of secondWord is "a" -> "0" -> 0.

The numerical value of targetWord is "aab" -> "001" -> 1.

We return false because 0 + 0 != 1.

Example 3:

Input: firstWord = "aaa", secondWord = "a", targetWord = "aaaa"

Output: true

Explanation:

The numerical value of firstWord is "aaa" -> "000" -> 0.

The numerical value of secondWord is "a" -> "0" -> 0.

The numerical value of targetWord is "aaaa" -> "0000" -> 0.

We return true because 0 + 0 == 0.

Constraints:

1 <= firstWord.length, secondWord.length, targetWord.length <= 8

firstWord, secondWord, and targetWord consist of lowercase English letters from 'a' to 'j' inclusive.

Ques - 4

Percentage of Letter in String

Given a string s and a character letter, return the percentage of characters in s that equal letter rounded down to the nearest whole percent.

Example 1:

Input: s = "foobar", letter = "o"

Output: 33

Explanation:

The percentage of characters in s that equal the letter 'o' is 2 / 6 * 100% = 33% when rounded down, so we return 33.

Example 2:

Input: s = "jjjj", letter = "k"

Output: 0

Explanation:

The percentage of characters in s that equal the letter 'k' is 0%, so we return 0.

Constraints:

1 <= s.length <= 100

s consists of lowercase English letters.

letter is a lowercase English letter

Ques - 5

Count the Number of Vowel Strings in Range

You are given a 0-indexed array of string words and two integers left and right.

A string is called a vowel string if it starts with a vowel character and ends with a vowel character where vowel characters are 'a', 'e', 'i', 'o', and 'u'.

Return the number of vowel strings words[i] where i belongs to the inclusive range [left, right].

Example 1:

Input: words = ["are","amy","u"], left = 0, right = 2

Output: 2

Explanation:

- "are" is a vowel string because it starts with 'a' and ends with 'e'.
- "amy" is not a vowel string because it does not end with a vowel.
- "u" is a vowel string because it starts with 'u' and ends with 'u'.

The number of vowel strings in the mentioned range is 2.

Example 2:

Input: words = ["hey","aeo","mu","ooo","artro"], left = 1, right = 4

Output: 3

Explanation:

- "aeo" is a vowel string because it starts with 'a' and ends with 'o'.
- "mu" is not a vowel string because it does not start with a vowel.
- "ooo" is a vowel string because it starts with 'o' and ends with 'o'.
- "artro" is a vowel string because it starts with 'a' and ends with 'o'.

The number of vowel strings in the mentioned range is 3.

Constraints:

1 <= words.length <= 1000

1 <= words[i].length <= 10

words[i] consists of only lowercase English letters.

0 <= left <= right < words.length

Day 12(Leet code 5 questions):

Ques - 1

Maximum Number of Words You Can Type

There is a malfunctioning keyboard where some letter keys do not work. All other keys on the keyboard work properly.

Given a string text of words separated by a single space (no leading or trailing spaces) and a string brokenLetters of all distinct letter keys that are broken, return the number of words in text you can fully type using this keyboard.

Example 1:

Input: text = "hello world", brokenLetters = "ad"

Output: 1

Explanation: We cannot type "world" because the 'd' key is broken.

Example 2:

Input: text = "leet code", brokenLetters = "lt"

Output: 1

Explanation: We cannot type "leet" because the 'l' and 't' keys are broken.

Example 3:

Input: text = "leet code", brokenLetters = "e"

Output: 0

Explanation: We cannot type either word because the 'e' key is broken.

Constraints:

1 <= text.length <= 104

0 <= brokenLetters.length <= 26

text consists of words separated by a single space without any leading or trailing spaces.

Each word only consists of lowercase English letters.

brokenLetters consists of distinct lowercase English letters.

Ques - 2

First Letter to Appear Twice

Given a string s consisting of lowercase English letters, return the first letter to appear twice.

Note:

A letter a appears twice before another letter b if the second occurrence of a is before the second occurrence of b.

s will contain at least one letter that appears twice.

Example 1:

Input: s = "abccbaacz"

Output: "c"

Explanation:

The letter 'a' appears on the indexes 0, 5 and 6.

The letter 'b' appears on the indexes 1 and 4.

The letter 'c' appears on the indexes 2, 3 and 7.

The letter 'z' appears on the index 8.

The letter 'c' is the first letter to appear twice, because out of all the letters the index of its second occurrence is the smallest.

Example 2:

Input: s = "abcdd"

Output: "d"

Explanation:

The only letter that appears twice is 'd' so we return 'd'.

Constraints:

2 <= s.length <= 100

s consists of lowercase English letters.

s has at least one repeated letter

Ques – 3

Split Strings by Separator

Given an array of strings words and a character separator, split each string in words by separator.

Return an array of strings containing the new strings formed after the splits, excluding empty strings.

Notes

separator is used to determine where the split should occur, but it is not included as part of the resulting strings.

A split may result in more than two strings.

The resulting strings must maintain the same order as they were initially given.

Example 1:

Input: words = ["one.two.three","four.five","six"], separator = "."

Output: ["one","two","three","four","five","six"]

Explanation: In this example we split as follows:

"one.two.three" splits into "one", "two", "three"

"four.five" splits into "four", "five"

"six" splits into "six"

Hence, the resulting array is ["one","two","three","four","five","six"].

Example 2:

Input: words = ["$easy$","$problem$"], separator = "$"

Output: ["easy","problem"]

Explanation: In this example we split as follows:

"$easy$" splits into "easy" (excluding empty strings)

"$problem$" splits into "problem" (excluding empty strings)

Hence, the resulting array is ["easy","problem"].

Example 3:

Input: words = ["|||"], separator = "|"

Output: []

Explanation: In this example the resulting split of "|||" will contain only empty strings, so we return an empty array [].

Constraints:

1 <= words.length <= 100

1 <= words[i].length <= 20

characters in words[i] are either lowercase English letters or characters from the string ".,|$#@" (excluding the quotes)

separator is a character from the string ".,|$#@" (excluding the quotes)

Ques - 4

Substrings of Size Three with Distinct Characters

A string is good if there are no repeated characters.

Given a string s, return the number of good substrings of length three in s.

Note that if there are multiple occurrences of the same substring, every occurrence should be counted.

A substring is a contiguous sequence of characters in a string.

Example 1:

Input: s = "xyzzaz"

Output: 1

Explanation: There are 4 substrings of size 3: "xyz", "yzz", "zza", and "zaz".

The only good substring of length 3 is "xyz".

Example 2:

Input: s = "aababcabc"

Output: 4

Explanation: There are 7 substrings of size 3: "aab", "aba", "bab", "abc", "bca", "cab", and "abc".

The good substrings are "abc", "bca", "cab", and "abc".

Constraints:

1 <= s.length <= 100

s consists of lowercase English letters.

Ques - 5

Kth Distinct String in an Array

A distinct string is a string that is present only once in an array.

Given an array of strings arr, and an integer k, return the kth distinct string present in arr. If there are fewer than k distinct strings, return an empty string "".

Note that the strings are considered in the order in which they appear in the array.

Example 1:

Input: arr = ["d","b","c","b","c","a"], k = 2

Output: "a"

Explanation:

The only distinct strings in arr are "d" and "a".

"d" appears 1st, so it is the 1st distinct string.

"a" appears 2nd, so it is the 2nd distinct string.

Since k == 2, "a" is returned.

Example 2:

Input: arr = ["aaa","aa","a"], k = 1

Output: "aaa"

Explanation:

All strings in arr are distinct, so the 1st string "aaa" is returned.

Example 3:

Input: arr = ["a","b","a"], k = 3

Output: ""

Explanation:

The only distinct string is "b". Since there are fewer than 3 distinct strings, we return an empty string "".

Constraints:

1 <= k <= arr.length <= 1000

1 <= arr[i].length <= 5

arr[i] consists of lowercase English letters.

Day 13(Leet code 5 questions):

Ques - 1

Count Prefixes of a Given String

You are given a string array words and a string s, where words[i] and s comprise only of lowercase English letters.

Return the number of strings in words that are a prefix of s.

A prefix of a string is a substring that occurs at the beginning of the string. A substring is a contiguous sequence of characters within a string.

Example 1:

Input: words = ["a","b","c","ab","bc","abc"], s = "abc"

Output: 3

Explanation:

The strings in words which are a prefix of s = "abc" are:

"a", "ab", and "abc".

Thus the number of strings in words which are a prefix of s is 3.

Example 2:

Input: words = ["a","a"], s = "aa"

Output: 2

Explanation:

Both of the strings are a prefix of s.

Note that the same string can occur multiple times in words, and it should be counted each time.

Constraints:

1 <= words.length <= 1000

1 <= words[i].length, s.length <= 10

words[i] and s consist of lowercase English letters only

Ques - 2

Fizz Buzz

Given an integer n, return a string array answer (1-indexed) where:

answer[i] == "FizzBuzz" if i is divisible by 3 and 5.

answer[i] == "Fizz" if i is divisible by 3.

answer[i] == "Buzz" if i is divisible by 5.

answer[i] == i (as a string) if none of the above conditions are true.

Example 1:

Input: n = 3

Output: ["1","2","Fizz"]

Example 2:

Input: n = 5

Output: ["1","2","Fizz","4","Buzz"]

Example 3:

Input: n = 15

Output:
["1","2","Fizz","4","Buzz","Fizz","7","8","Fizz","Buzz","11","Fizz","13","14","FizzBuzz"]

Constraints:

1 <= n <= 104

Ques - 3

Check if Number Has Equal Digit Count and Digit Value

You are given a 0-indexed string num of length n consisting of digits.

Return true if for every index i in the range 0 <= i < n, the digit i occurs num[i] times in num, otherwise return false.

Example 1:

Input: num = "1210"

Output: true

Explanation:

num[0] = '1'. The digit 0 occurs once in num.

num[1] = '2'. The digit 1 occurs twice in num.

num[2] = '1'. The digit 2 occurs once in num.

num[3] = '0'. The digit 3 occurs zero times in num.

The condition holds true for every index in "1210", so return true.

Example 2:

Input: num = "030"

Output: false

Explanation:

num[0] = '0'. The digit 0 should occur zero times, but actually occurs twice in num.

num[1] = '3'. The digit 1 should occur three times, but actually occurs zero times in num.

num[2] = '0'. The digit 2 occurs zero times in num.

The indices 0 and 1 both violate the condition, so return false.

Constraints:

n == num.length

1 <= n <= 10

num consists of digits.

Ques - 4

Maximum Value of a String in an Array

The value of an alphanumeric string can be defined as:

The numeric representation of the string in base 10, if it comprises of digits only.

The length of the string, otherwise.

Given an array strs of alphanumeric strings, return the maximum value of any string in strs.

Example 1:

Input: strs = ["alic3","bob","3","4","00000"]

Output: 5

Explanation:

- "alic3" consists of both letters and digits, so its value is its length, i.e. 5.
- "bob" consists only of letters, so its value is also its length, i.e. 3.
- "3" consists only of digits, so its value is its numeric equivalent, i.e. 3.
- "4" also consists only of digits, so its value is 4.
- "00000" consists only of digits, so its value is 0.

Hence, the maximum value is 5, of "alic3".

Example 2:

Input: strs = ["1","01","001","0001"]

Output: 1

Explanation:

Each string in the array has value 1. Hence, we return 1.

Constraints:

1 <= strs.length <= 100

1 <= strs[i].length <= 9

strs[i] consists of only lowercase English letters and digits

Ques - 5

Shortest Distance to a Character

Given a string s and a character c that occurs in s, return an array of integers answer where answer.length == s.length and answer[i] is the distance from index i to the closest occurrence of character c in s.

The distance between two indices i and j is abs(i - j), where abs is the absolute value function.

Example 1:

Input: s = "loveleetcode", c = "e"

Output: [3,2,1,0,1,0,0,1,2,2,1,0]

Explanation: The character 'e' appears at indices 3, 5, 6, and 11 (0-indexed).

The closest occurrence of 'e' for index 0 is at index 3, so the distance is abs(0 - 3) = 3.

The closest occurrence of 'e' for index 1 is at index 3, so the distance is abs(1 - 3) = 2.

For index 4, there is a tie between the 'e' at index 3 and the 'e' at index 5, but the distance is still the same: abs(4 - 3) == abs(4 - 5) = 1.

The closest occurrence of 'e' for index 8 is at index 6, so the distance is abs(8 - 6) = 2.

Example 2:

Input: s = "aaab", c = "b"

Output: [3,2,1,0]

Constraints:

1 <= s.length <= 104

s[i] and c are lowercase English letters.

It is guaranteed that c occurs at least once in s.

Day 14(Leet code 5 questions):

Ques - 1

Check if All A's Appears Before All B's

Given a string s consisting of only the characters 'a' and 'b', return true if every 'a' appears before every 'b' in the string. Otherwise, return false.

Example 1:

Input: s = "aaabbb"

Output: true

Explanation:

The 'a's are at indices 0, 1, and 2, while the 'b's are at indices 3, 4, and 5.

Hence, every 'a' appears before every 'b' and we return true.

Example 2:

Input: s = "abab"

Output: false

Explanation:

There is an 'a' at index 2 and a 'b' at index 1.

Hence, not every 'a' appears before every 'b' and we return false.

Example 3:

Input: s = "bbb"

Output: true

Explanation:

There are no 'a's, hence, every 'a' appears before every 'b' and we return true.

Constraints:

1 <= s.length <= 100

s[i] is either 'a' or 'b'

Ques - 2

Count Common Words With One Occurrence

Given two string arrays words1 and words2, return the number of strings that appear exactly once in each of the two arrays.

Example 1:

Input: words1 = ["leetcode","is","amazing","as","is"], words2 = ["amazing","leetcode","is"]

Output: 2

Explanation:

- "leetcode" appears exactly once in each of the two arrays. We count this string.

- "amazing" appears exactly once in each of the two arrays. We count this string.

- "is" appears in each of the two arrays, but there are 2 occurrences of it in words1. We do not count this string.

- "as" appears once in words1, but does not appear in words2. We do not count this string.

Thus, there are 2 strings that appear exactly once in each of the two arrays.

Example 2:

Input: words1 = ["b","bb","bbb"], words2 = ["a","aa","aaa"]

Output: 0

Explanation: There are no strings that appear in each of the two arrays.

Example 3:

Input: words1 = ["a","ab"], words2 = ["a","a","a","ab"]

Output: 1

Explanation: The only string that appears exactly once in each of the two arrays is "ab".

Constraints:

1 <= words1.length, words2.length <= 1000

1 <= words1[i].length, words2[j].length <= 30

words1[i] and words2[j] consists only of lowercase English letters.

Ques - 3

Greatest English Letter in Upper and Lower Case

Given a string of English letters s, return the greatest English letter which occurs as both a lowercase and uppercase letter in s. The returned letter should be in uppercase. If no such letter exists, return an empty string.

An English letter b is greater than another letter a if b appears after a in the English alphabet.

Example 1:

Input: s = "lEeTcOdE"

Output: "E"

Explanation:

The letter 'E' is the only letter to appear in both lower and upper case.

Example 2:

Input: s = "arRAzFif"

Output: "R"

Explanation:

The letter 'R' is the greatest letter to appear in both lower and upper case.

Note that 'A' and 'F' also appear in both lower and upper case, but 'R' is greater than 'F' or 'A'.

Example 3:

Input: s = "AbCdEfGhIjK"

Output: ""

Explanation:

There is no letter that appears in both lower and upper case.

Constraints:

1 <= s.length <= 1000

s consists of lowercase and uppercase English letters.

Ques - 4

Count Pairs Of Similar Strings

You are given a 0-indexed string array words.

Two strings are similar if they consist of the same characters.

For example, "abca" and "cba" are similar since both consist of characters 'a', 'b', and 'c'.

However, "abacba" and "bcfd" are not similar since they do not consist of the same characters.

Return the number of pairs (i, j) such that 0 <= i < j <= word.length - 1 and the two strings words[i] and words[j] are similar.

Example 1:

Input: words = ["aba","aabb","abcd","bac","aabc"]

Output: 2

Explanation: There are 2 pairs that satisfy the conditions:

- i = 0 and j = 1 : both words[0] and words[1] only consist of characters 'a' and 'b'.

- i = 3 and j = 4 : both words[3] and words[4] only consist of characters 'a', 'b', and 'c'.

Example 2:

Input: words = ["aabb","ab","ba"]

Output: 3

Explanation: There are 3 pairs that satisfy the conditions:

- i = 0 and j = 1 : both words[0] and words[1] only consist of characters 'a' and 'b'.

- i = 0 and j = 2 : both words[0] and words[2] only consist of characters 'a' and 'b'.

- i = 1 and j = 2 : both words[1] and words[2] only consist of characters 'a' and 'b'.

Example 3:

Input: words = ["nba","cba","dba"]

Output: 0

Explanation: Since there does not exist any pair that satisfies the conditions, we return 0.

Constraints:

1 <= words.length <= 100

1 <= words[i].length <= 100

words[i] consist of only lowercase English letters

Ques - 5

Minimum String Length After Removing Substrings

You are given a string s consisting only of uppercase English letters.

You can apply some operations to this string where, in one operation, you can remove any occurrence of one of the substrings "AB" or "CD" from s.

Return the minimum possible length of the resulting string that you can obtain.

Note that the string concatenates after removing the substring and could produce new "AB" or "CD" substrings.

Example 1:

Input: s = "ABFCACDB"

Output: 2

Explanation: We can do the following operations:

- Remove the substring "ABFCACDB", so s = "FCACDB".
- Remove the substring "FCACDB", so s = "FCAB".
- Remove the substring "FCAB", so s = "FC".

So the resulting length of the string is 2.

It can be shown that it is the minimum length that we can obtain.

Example 2:

Input: s = "ACBBD"

Output: 5

Explanation: We cannot do any operations on the string so the length remains the same.

Constraints:

1 <= s.length <= 100

s consists only of uppercase English letters

2D Array (14 Questions)

Day 1(9 Questions):

Ques - 1

Write a java program to create a 2D array of 3x5 size and store elements by reading from user , and also store element directly.

Ques - 2

Write a java program to search a given numbers location from a 2D array.

```
Input : mat[][] = { {5, 4, 7},
                    {1, 3, 8},
                    {2, 9, 6} }
```

Enter element to be searched : 3

Output : 3 is present at (1,1) location

Ques - 3

Write a java program to find the sum of all elements from a given 2D array.

```
Input : mat[][] = { {5, 4, 7},
                    {1, 3, 8},
                    {2, 9, 6} }
```

Output : Sum of all element is : 46

Ques - 4

Write a java program to print the max number of each rows from a 2D array.

```
Input : mat[][] = { {5, 4, 7},
                  {1, 3, 8},
                  {2, 9, 6} }
Output : max number of 1 row : 7
        max number of 2 row : 8
        max number of 3 row : 9
```

Ques - 5

Sort the given matrix

Given a n x n matrix. The problem is to sort the given matrix in strict order. Here strict order means that the matrix is sorted in a way such that all elements in a row are sorted in increasing order and for row ‘i’, where 1 <= i <= n-1, the first element of row ‘i’ is greater than or equal to the last element of row ‘i-1’.

Examples:

```
Input : mat[][] = { {5, 4, 7},
                  {1, 3, 8},
                  {2, 9, 6} }
Output :
        1 2 3
        4 5 6
        7 8 9
```

Ques – 6

Count all sorted rows in a matrix

Given a matrix of m*n size, the task is to count all the rows in a matrix that are sorted either in strictly increasing order or in strictly decreasing order?

Examples :

```
Input : m = 4,  n = 5
       mat[m][n] =
1 2 3 4 5
4 3 1 2 6
8 7 6 5 4
5 7 8 9 10
```

Output: 3

Ques - 7

Efficiently compute sums of diagonals of a matrix

Given a 2D square matrix, find the sum of elements in Principal and Secondary diagonals. For example, consider the following 4 X 4 input matrix.

Input :

```
4

1       2     3     4

4       3     2    1

7       8     9    6

6       5     4    3
```

Output :

Principal Diagonal: 16

Secondary Diagonal: 20

Input :

```
3
1 1 1
1 1 1
1 1 1
```

Output :

Principal Diagonal: 3

Secondary Diagonal: 3

Ques - 8

Printing Boundary Elements of a Matrix.

Given a matrix of size n x m. Print the boundary elements of the matrix. Boundary elements are those elements that are not surrounded by elements in all four directions, i.e. elements in the first row, first column, last row, and last column

Examples:

Input:

```
1 2 3 4
5 6 7 8
1 2 3 4
5 6 7 8
```

Output :

```
1 2 3 4
5     8
1     4
5 6 7 8
```

Input:

```
1 2 3
5 6 7
1 2 3
```

Output:

```
1 2 3
```

```
    5   7
    1 2 3
```

Ques - 9

Print all elements in sorted order from row and column wise sorted matrix by using Collection.

```
nput: mat[][] = { {10, 20, 30, 40},
                  {15, 25, 35, 45},
                  {27, 29, 37, 48},
                 {32, 33, 39, 50},
             };
```

Output: 10 15 20 25 27 29 30 32 33 35 37 39 40 45 48 50

[Note: consider all elements are unique only]

Day 2(5 Questions):

Ques - 1

Java Program to find Sum of all elements of each row of a matrix.

Input :

```
1 2 3
4 5 6
7 8 9
```

Output :

```
1 2 3  6
4 5 6  15
7 8 9  24
```

Ques - 2

Java Program to print lower diagonal of a matrix.

Input :

```
1         2          3
4         5          6
7         8          9
```

Output :

```
1 0 0
4 5 0
7 8 9
```

Ques - 3

Java program to check a given matrix is an identity matrix or not.

The identity matrix of size n is the n x n square matrix with ones on the main diagonal and zeros elsewhere.

Input :

```
int Matrix1[3][3] = {
        { 1, 0, 0 },
        { 0, 1, 0 },
        { 0, 0, 1 }
    };
int Matrix2[3][3] = {
        { 1, 0, 0 },
        { 2, 1, 0 },
        { 0, 0, 1 }
    };
```

Output:

```
Matrix1 is an IDENTITY MATRIX
Matrix2 is NOT an identity matrix
```

Ques - 4

Java program to check a given matrix is a sparse matrix or not.

```
RUN 1:
Enter the elements of the matrix:
4 5 6
7 0 0
0 0 0
Matrix is a Sparse Matrix

RUN 2:
Enter the elements of the matrix:
1 2 3
4 5 6
7 8 9
Matrix is not a Sparse Matrix
```

Ques - 5

Java program to print the upper triangular matrix.

```
Matrix:
9 8 7
5 4 6
1 2 3

Upper triangular matrix is:
9 8 7
  4 6
    3
```

Object Array(8 Questions)

Day 1(6 Questions):

(Create a student class with two instance variable id and name

which are private variables, develop a constructor to initialize the instance varibales and override toString method.)

1)Develop a java to program to store Students in a Student array and count how many duplicate students ids are there based on their id.

2)Develop a java to program to store Students in a Student array and count how many duplicate students names are there based on their name.

3)Develop a java to program to store Students in a Student array and count how many duplicate students(Based on name and id) are there by comparing two object.

Ques – 4

Finding Duplicate Student IDs

You are given an array of Student objects. Each Student has attributes name, age, and id. Write a Java program to find and print duplicate student IDs.

Your task is to implement the findDuplicateIds method which takes an array of Student objects and prints the IDs of the students that have duplicates.

Instructions:

Implement the findDuplicateIds method in the Main class.

Inside the findDuplicateIds method, iterate through the array of Student objects and print the IDs of the students that have duplicates.

If no duplicates are found, print "No duplicates found."

You may assume that the Student class is already provided to you.

The Student class has attributes name, age, and id, along with overridden equals and hashCode methods for comparison based on the id attribute.

You should not modify the Student class.

Ques - 5)

Unique Product Array in Java

Description:

You are tasked with implementing a Java program to manage a array of products, ensuring that only unique products are stored in an array. Your program should consist of two classes: Product and ProductArray.

The Product class should have the following attributes:

id (integer): Represents the unique identifier of the product.

name (string): Represents the name of the product.

price (double): Represents the price of the product.

The ProductArray class should manage an array of products with a maximum capacity of 10. It should provide the following functionalities:

An array to store products.

A method addProduct(Product product) to add a product to the array. Ensure that duplicate products (products with the same id) are not added.

A method printDetails() to print the details of all the unique products stored in the array.

Ensure that your implementation follows good programming practices, including encapsulation, proper naming conventions, and commenting where necessary.

Your task is to implement the Product and ProductArray classes according to the specifications provided above and demonstrate their usage in the main method by adding products to the array and printing their details.

Ques - 6

Imagine you're developing a student management system for a school. You need to implement a class called StudentManager that allows users to perform various operations on student objects. The class should have methods to add a student, retrieve a student by their ID, retrieve a student object by index, remove a student, clear the array of students, and check if a student exists in the system.

Add Student: Implement a method addStudent() that takes in a student object and adds it to the array of students.

Get Student by ID: Create a method getStudentById() which takes a student ID as input and returns the corresponding student object if it exists, otherwise returns null.

Get Student by Index: Develop a method getStudentByIndex() that takes an index as input and returns the student object at that index if it exists, otherwise returns null.

Remove Student: Implement a method removeStudent() that takes a student ID as input and removes the corresponding student from the list if found. If the student doesn't exist, it should return false, otherwise true.

Clear Students: Create a method clearStudents() that removes all students from the list.

Contains Student: Add a method containsStudent() that takes a student ID as input and returns true if the student exists in the system, otherwise false.

Your task is to implement these methods in the StudentManager class and demonstrate their functionality with a sample usage scenario. Provide sample code showing how you would use each method to add, retrieve, remove, clear, and check for the existence of students in the system. Also, include appropriate error handling where necessary.

Day 2(2 Questions):

Ques - 1

Imagine you're developing a student management system for a school. You need to implement a class called StudentManager that allows users to perform various operations on student objects. The class should have methods to add a student, retrieve a student by their ID, retrieve a student object by index, remove a student, clear the array of students, and check if a student exists in the system.

Add Student: Implement a method addStudent() that takes in a student object and adds it to the array of students.

Get Student by ID: Create a method getStudentById() which takes a student ID as input and returns the corresponding student object if it exists, otherwise returns null.

Get Student by Index: Develop a method getStudentByIndex() that takes an index as input and returns the student object at that index if it exists, otherwise returns null.

Remove Student: Implement a method removeStudent() that takes a student ID as input and removes the corresponding student from the list if found. If the student doesn't exist, it should return false, otherwise true.

Clear Students: Create a method clearStudents() that removes all students from the array.

Contains Student: Add a method containsStudent() that takes a student ID as input and returns true if the student exists in the system, otherwise false.

Your task is to implement these methods in the StudentManager class and demonstrate their functionality with a sample usage scenario. Provide sample code showing how you would use each method to add, retrieve, remove, clear, and check for the existence of students in the system. Also, include appropriate error handling where necessary.

Ques - 2

Employee Management System with Index-Based Retrieval

Description:

You are tasked with implementing an Employee Management System in Java that allows users to manage employees effectively. The system should enable users to add, retrieve by index, remove, clear, check for existence, and get the size of the employee list.

Here are the requirements:

Implement the Employee class with the following properties:

employeeId (int): Unique identifier for the employee.

name (String): Name of the employee.

position (String): Position or job title of the employee.

Provide appropriate constructor and getter methods for the properties.

Implement the EmployeeManager class to manage employees. It should have the following methods:

addEmployee(Employee employee): Add an employee to the array.

getEmployee(int index): Retrieve an employee from the array based on the provided index.

removeEmployee(Employee employeeToRemove): Remove an employee from the system based on the provided employee object.

clearEmployees(): Remove all employees from the array.

containsEmployee(Employee employeeToCheck): Check if the array contains the provided employee object.

isEmpty(): Check if the employee array is empty.

Ensure that the equals method is properly overridden in the Employee class to compare two employee objects based on their properties.

Provide a sample usage scenario in the main method of the EmployeeManager class to demonstrate the functionality of the system. Include adding employees, retrieving employees by index, removing employees, clearing the employee list, checking for employee existence, and checking if the employee list is empty.

Collection (25 Programs)

1)Write a program to convert a given array into a collection with the asList() method.

String[] str = {"Abc","Mno","Prq"};

2)How to Make a Collection Read-Only in Java?

List = [1,2,3,4,5,6]

1.Read only List object

2.Read only Set object

[Read only means modification is not allowed]

3)Java Program to Compare Elements in a Collection.

Input : List = [3, 5, 18, 4, 6]

Output: Min value of our list : 3

Max value of our list : 18

Input : List = [4,4,4,4,4]

Output : All elements are same.

4)Java Program to Remove a Specific Element all occurence From a Collection.

List = [1,2,2,3,4,2,5,2]

element : 2

List = [1,3,4,5]

5)How to remove an element from ArrayList in Java?

6)How to Replace an Element in Java ArrayList?

1.Replace one old element to new element

List = [1,2,3,2,4,5]

oldEle : 2

newEle : 20

List = [1,20,3,2,4,5]

2.Replace all the ocurrence of old ele to new ele.

List = [1,2,3,2,4,5]

oldEle : 2

newEle : 20

List = [1,20,3,20,4,5]

7)Compare two ArrayList In Java?

List1 = [1,2,3]

List2 = [1,2,3]

Output : true

List1 = [6,2,3]

List2 = [1,2,3]

Output : false

8)Find the prime integers from a List of Integers and print it.

List = [1,2,3,4,5]

Output : 2 3 5

9)Java Program to Convert ArrayList to LinkedList.

10)Java Program to Convert Array to LinkedList in Java.

11)How to Sort HashSet in Java.

12)Convert Array to HashSet in Java.

13)Store n times "Java" in collection object.

n -> 5

List = ["Java", "Java", "Java", "Java"]

14)Find how many times a given number appeared in list and print the count.

List = [1,2,1,2,3,1,3,1]

Count of 1 : 4

-> Print all the occurence of each element of the above list.

1 : 4 times

2 : 2 times

3 : 3 times

15)Check the two collection objects contains unique elements.

List = [1,2,3,4]

Set = [6,7,8,9]

Output : They both contains different objects

16)Rotate ArrayList elements n times.

List = [1,2,3,4,5,6]

Rotate : 4

Output : [4, 5, 6, 1, 2, 3]

17)Swap two index element in a collection object.

List = [1,2,3,4,5,6]

index1 = 2

index2 = 4

Output : [1,2,5,4,3,6]

18)Reverse the elements in collection object.

19)Print all the even and odd elements separately from a List of Integers.

20)Print only common integers from two ArrayList of Integers.

List1 = [1,2,3,4,5]

List2 = [1,2,6,7,3]

Output : [1, 2, 3]

21)Print only uncommon integers from two ArrayList of Integers.

List1 = [1,2,3,4,5]

List2 = [1,2,6,7,3]

Output: [5,6,7]

22) Check the List is having at least one same element or all different elements.

List1 = [1,2,3,4,5]

List2 = [1,2,6,7,3]

Output : Two list is having atleast one same element

List1 = [1,2,3,4,5]

List2 = [12,13,14,15]

Output : Two list is having all different elements.

23)Given a list of strings, write a Java program to filter out and print all strings that contain the letter "a".

24)Write a program to convert all the strings in a list to uppercase and print the result.

25)Given two lists of integers, write a program to combine them into a single list and then find and print the sum of all elements in the combined list.