SRMIST
Faculty of Engineering and Technology
School of Computing
Data Structures and Algorithms Lab
Unit – 1

1. Given a limited range of array of size 'n' containing elements between 1 to n-1 with one element repeating, find the duplicate number in it without using any extra space. The output must be displayed along with the location of the duplicate element.

2. Merge two sorted arrays array1 [u] and array2[v], the array elements should remain as sorted. The size of array1 is sufficient enough to hold both array values. Try to solve the problem using brute force technique.

3. Find the missing numbers in an array and also check whether the number that's missing is even or odd.

4. Print the largest and smallest element in an unsorted array.

5. Given an array containing only 0's, 1's, and 2's, sort it in linear time and using constant space.

6. The program must accept a string S as input. The program is to be designed to print the second occurring vowel in the string S as the output otherwise an appropriate message.

7. Design a program such that N integers are accepted as input and displays the number of occurrence of last digit in the given integer.

8. The program must accept two lines of integers as input, first line of integer is in sorting order. Print the count of integers in second line which appears in first line as output.

9. You are given an array A with size N and two integers M and K.
    Let's define another array B with size N×M as the array that's formed by concatenating M copies of array A.
    You have to find the number of sub-arrays of the array B with sum ≤K. Since the answer can be very large you have to print the answer mod 10^9+7.

    Input Format:

    The first line contains an integer T denoting the number of test cases.
    The first line of each test case contains 3 space separated integers N and M and K.
    Next line contains N space separated integers denoting the array elements.

    Output Format:

    For each test case, print the required answer in a new line.

    Constraints:
    $1 \leq T \leq 10$
    $1 \leq N, M \leq 10^5$
    $1 \leq K \leq 10^{16}$
    $1 \leq Ai \leq 10^6$

10. There are N warehouses. The warehouses are located in a straight line and are indexed from 1 to N. Each warehouse contains some number of sacks.

A thief decides to rob these warehouses. Thief figured out that he can escape the police if and only if he follows both the following 2 constraints:

He will rob only one continuous segment of warehouses.

He will rob same number of sacks from each warehouse.

Thief wants to calculate the maximum number of sacks he can steal without getting caught by the police.

Input Format:

The first line contains an integer T denoting number test cases.

The first line of each test case contains a single integer N denoting number of warehouses.

The second line of each test case contains N space-separated integers :a[1],a[2],a[3]...a[n].a[i] denotes number of sacks in ith warehouse.

Constraints:

$1 <= T <= 5$

$1 <= N <= 10^6$

$0 <= A[i] <= 10^{12}$

11. A contiguous subarray is defined as unique if all the integers contained within it occur exactly once. There is a unique weight associated with each of the subarray. Unique weight for any subarray equals it's length if it's unique, 0 otherwise. Your task is to calculate the sum of unique weights of all the contiguous subarrays contained within a given array.

Input
First line of the input contains an integer T, denoting the number of testcases.
2*T lines follow, where first line of each testcase contains an integer N denoting the number of integers in the given array. Last line of each testcase then contains N single space separated integers

Output
Print the summation of unique weights of all the subarrays for each testcase in a separate line.

Constraints

- $1 \leq T, N \leq 10^5$

- $0 \le A_i \le 10^9$
- Summation of N for all T does not exceed 10^5

12. A bracket sequence is a string that contains only characters '(' and ')'.

A correct bracket sequence is a bracket sequence that can be transformed into a correct arithmetic expression by inserting characters '1' and '+' between the original characters of the sequence. For example, bracket sequences '()()' and '(())' are correct. The resulting expressions of these sequences are: '(1)+(1)' and '((1+1)+1)'. However, '(', ')(', and '(' are incorrect bracket sequences.

You are given a bracket sequence s(s1s2...sn), where si denotes the type of i's bracket (open or close). It is not mandatory that s is necessarily correct. Your task is to determine the number of i's such that si si+1...sn s1 s2...si−1 is a correct bracket sequence.

Input format

The single line contains sequence s.

Output format

1. Print the number of shifts denoting the correct bracket sequence

13. Given an array **arr[]** and an integer **K** where K is smaller than size of array, the task is to find the **Kth smallest** element in the given array. It is given that all array elements are distinct.
Input: swize of array = 5
Array [] = 7 10 4 5 2
Kth element = 4
Output = 7 (since 4th smallest element in the array is 7)

14. Given an unsorted array A of size N that contains only non-negative integers, find a continuous sub-array which adds to a given number S.
```
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.
```
15. Given an unsorted array **arr[]** of size **N** having both negative and positive integers. The task is place all negative element at the end of

array without changing the order of positive element and negative element.

Input :
N = 8
arr[] = {1, -1, 3, 2, -7, -5, 11, 6 }
Output :
1 3 2 11 6 -1 -7 -5

All Solutions must contain:
    a. Provide follow-up questions
    b. Dry run
    c. Complexity
    d. Validation of input
    e. Proper code
    f. Result in the proper format