Self-Practice No. : **1**

Topics Covered : **Basic Math, Control Flow, Arrays, Functions, Bit Manipulation**

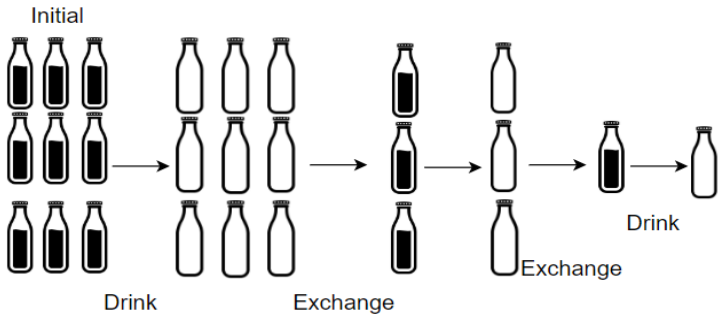
Date : **08-05-2024**

Solve the following problems

Q No.	Question Detail	Level
1	**Calculate Delayed Arrival Time** **Problem statement:** You are given a positive integer arrivalTime denoting the arrival time of a train in hours, and another positive integer delayedTime denoting the amount of delay in hours. Return the time when the train will arrive at the station. Note that the time in this problem is in 24-hours format. **Example 1:** **Input:** arrivalTime = 15, delayedTime = 5 **Output:** 20 **Explanation:** Arrival time of the train was 15:00 hours. It is delayed by 5 hours. Now it will reach at 15+5 = 20 (20:00 hours). **Example 2:** **Input:** arrivalTime = 13, delayedTime = 11 **Output:** 0 **Explanation:** Arrival time of the train was 13:00 hours. It is delayed by 11 hours. Now it will reach at 13+11=24 (Which is denoted by 00:00 in 24 hours format so return 0). **Constraints:** • 1 <= arrivaltime < 24 • 1 <= delayedTime <= 24	Easy

2	**Recycling Pens** **Problem statement:** You have 'N' empty pens whose refills have been used up. You have 'R' rupees in your pocket. You have two choices of operations that you can perform each time. 1) Recycle 1 empty pen and get 'K' rupees as a reward. 2) Buy 1 refill for 'C' rupees and combine it with 1 empty pen to make one usable pen. Your task is to find the maximum number of usable pens that you can get. For example if you have 'N' = 5 , 'R' = 10 , 'K' = 2 , 'C' = 3. You can recycle one pen and get 2 rupees as a reward so you will have a total of 12 rupees. Now you can buy 4 refills and combine it with 4 pens to make it usable. So your answer is 4. **Sample Input 1 :** 3 10 10 5 5 15 11 3 5 3 20 20 2 **Sample Output 1 :** 6 7 3 **Explanation for Sample input 1** : In the first test case, you will sell 4 empty pens and you will get 20 rupees so your total money will be 10+20 = 30 and from that, you will buy 6 refills and make 6 usable pens. In second test case you will sell 8 pens and you will get 8*3=24 rupees and your total money will be 24+11 = 35 and from that, you will buy 7 refills and make 7 usable pens. In this test case, you have a lot of money to buy but you have only 3 empty pens so you buy 3 refills for 6 rupees and make 3 usable pens. **Sample Input 2 :**	

	3 10 10 1 10 5 0 5 5 6 0 4 2 **Sample Output 2 :** 1 2 4 **Explanation of Sample Input 2 :** In the first test case you can buy 1 refill from the money you have and make 1 usable pen. In the second test case you can sell 2 empty pens and get 10 rupees and from it, you can buy 2 refills and make 2 usable pens. In the third test case you can sell 2 empty pens and get 8 rupees and from it, you can buy 4 refills and make 4 usable pens. **Constraints :** 1 <= T <= 10^5 1 <= N <= 10^9 0 <= R <= 10^9 1 <= K <= 10^9 1 <= C <= 10^9	
3	**Water Bottles** **Problem statement:** There are numBottles water bottles that are initially full of water. You can exchange numExchange empty water bottles from the market with one full water bottle. The operation of drinking a full water bottle turns it into an empty bottle. Given the two integers numBottles and numExchange, return the maximum number of water bottles you can drink. **Example 1:**	



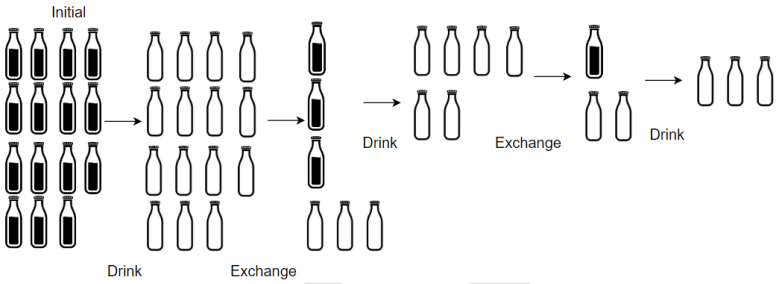
Input: numBottles = 9, numExchange = 3

Output: 13

Explanation: You can exchange 3 empty bottles to get 1 full water bottle.

Number of water bottles you can drink: 9 + 3 + 1 = 13.

Example 2:



Input: numBottles = 15, numExchange = 4

Output: 19

Explanation: You can exchange 4 empty bottles to get 1 full water bottle.

Number of water bottles you can drink: 15 + 3 + 1 = 19.

Constraints:

1 <= numBottles <= 100

2 <= numExchange <= 100

4 **Square Hollow pattern**

Problem statement : Print a square hollow pattern of asterisks (''). The pattern should consist of a hollow square with '' characters forming the outer boundary and spaces (' ') forming the inner region. The size of the square should be

	determined by the user input. The program should prompt the user to enter the size of the square and then print the corresponding hollow square pattern. **Input 1:** Size of square: 5 **Output 1:** ***** * * * * * * *****	
5	**Reverse Bits** **Problem Statement:** Given a 32 bit number x, reverse its binary form and return the answer in decimal. **Example 1:** **Input:** x = 1 **Output:** 2147483648 **Explanation:** Binary of 1 in 32 bits representation- 00000000000000000000000000000001 Reversing the binary form we get, 10000000000000000000000000000000, whose decimal value is 2147483648. **Example 2:** **Input:** x = 5 **Output:** 2684354560 **Explanation:** Binary of 5 in 32 bits representation- 00000000000000000000000000000101	

	Reversing the binary form we get, 10100000000000000000000000000000, whose decimal value is 2684354560. **Constraints:** • 1<= x < 2^32	
6	**XOR Operation in an Array** **Problem Statement:** You are given an integer n and an integer start. Define an array nums where nums[i] = start + 2 * i (0-indexed) and n == nums.length. Return the bitwise XOR of all elements of nums. **Example 1**: **Input:** n = 5, start = 0 **Output:** 8 **Explanation:** Array nums is equal to [0, 2, 4, 6, 8] where (0 ^ 2 ^ 4 ^ 6 ^ 8) = 8. Where "^" corresponds to bitwise XOR operator. **Example 2:** **Input:** n = 4, start = 3 **Output:** 8 **Explanation:** Array nums is equal to [3, 5, 7, 9] where (3 ^ 5 ^ 7 ^ 9) = 8. **Constraints:** • 1 <= n <= 1000 • 0 <= start <= 1000 • n == nums. Length	
7	**Party of Couple** **Problem Statement:** You are given an integer array arr[] of size n, representing n number of people in a party, each person is denoted by an integer. Couples are represented by the same number ie: two people have the same integer value, it means they are a couple. Find out the only single person in the party of couples. **Example 1:**	

	Input: n = 5 arr = {1, 2, 3, 2, 1} **Output:** 3 **Explanation:** Only the number 3 is single. **Example 2:** **Input:** n = 11 arr = {1, 2, 3, 5, 3, 2, 1, 4, 5, 6, 6} **Output:** 4 **Explanation**: 4 is the only single. **Constraints:** • 1 ≤ n ≤ 10^4 • 1 ≤ arr[i] ≤ 10^6	
8	**Transpose Matrix** **Problem Statement :** Given a 2D integer array matrix, return the transpose of matrix. The transpose of a matrix is the matrix flipped over its main diagonal, switching the matrix's row and column indices. **Example 1:** **Input**: matrix = [[1,2,3],[4,5,6],[7,8,9]] **Output**: [[1,4,7],[2,5,8],[3,6,9]] **Example 2:** **Input:** matrix = [[1,2,3],[4,5,6]] **Output**: [[1,4],[2,5],[3,6]] **Constraints:** m == matrix.length n == matrix[i].length 1 <= m, n <= 1000	

	1 <= m * n <= 10^5 -10^9 <= matrix[i][j] <= 10^9	
9	**Valid Mountain Array** **Problem statement** : Given an array of integers arr, return true if and only if it is a valid mountain array. Recall that arr is a mountain array if and only if: • arr.length >= 3 • There exists some i with 0 < i < arr.length - 1 such that: • arr[0] < arr[1] < ... < arr[i - 1] < arr[i] • arr[i] > arr[i + 1] > ... > arr[arr.length - 1] **Example 1:** **Input:** arr = [2,1] **Output:** false **Example 2:** **Input:** arr = [3,5,5] **Output:** false **Example 3:** **Input:** arr = [0,3,2,1] **Output:** true **Constraints:** 1 <= arr.length <= 10^4 0 <= arr[i] <= 10^4	
10	**Find Odd Occurrence Element** **Problem statement** : You are given an array of 'N' elements. In this given array, each element appears an even number of times except one element which appears odd no. of times. Your task is to find the element which occurs an odd number of times. **For example :**	

Input array [5,5,6,4,6],If we look at the frequency of different elements in this array.We can see,4 appears an odd number of times, so our answer will be 4.

Sample input 1 :

4
5
2 7 7 7 2
3
9 9 9
7
1 2 3 3 2 1 4
5
11 20 5 5 20

Sample output 1 :

7
9
4
11

Explanation for sample output 1 :

(i) For the first array, element 7 appears 3 times.
(ii) For the second array, element 9 appears 3 times.
(iii) For the third array, element 4 appears 1 time.
(iv) For the fourth array, element 11 appears 1 time.

Sample input 2 :

5
5
1 1 2 2 1
3
9 2 9
7
9 7 6 9 9 9 7
9
6 6 4 6 5 5 4 6 6
1
10

Sample output 2 :

1
2
6
6
10

Explanation for sample output 2:

(i) For the first array, element 1 appears 3 times.

(ii) For the second array, element 2 appears 1 time.

(iii) For the third array, element 6 appears 1 time.

(iv) For the fourth array, element 6 appears 5 times.

(v) For the fifth array, there is only 1 element, so the answer is 10.

Constraints:

1<= T <=100

1 <= N <= 10000

1 <= ARR[i] <= 10^8

Where 'T' denotes the number of test cases, 'N' denotes the number of elements in the array, and 'ARR[i]' denotes the 'i'th' element of the array 'ARR'.