VIGNAN'S
Foundation for Science, Technology & Research
(Deemed to be University)
-Estd. u/s 3 of UGC Act 1956

CODING CONTEST

1)Finding Missing Number

Given an array of size N, it contains all the numbers from 1 to N+1 inclusive, except one number. You have to find the missing number.

Input Format

First line of input contains T - number of test cases. Its followed by 2T lines, first line of each test case contains N - size of the array and the next line contains N integers - the elements of the array.

Constraints

1 <= T <= 500

1 <= N <= 10000

1 <= ar[i] <= N+1

Output Format

For each test case, print the missing number, separated by newline.

Sample Input 0

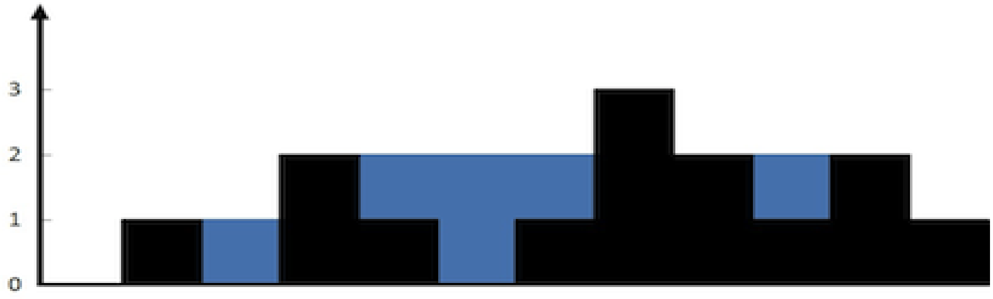
```
3
8
1 2 7 9 5 6 3 8
7
3 5 8 1 4 7 2
10
8 11 10 2 7 4 3 5 1 6
```

Sample Output 0

```
4
6
9
```

2)Trapping rain water

Given n non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it can trap after raining.



Input: height = [0,1,0,2,1,0,1,3,2,1,2,1]

Output: 6

Explanation: The above elevation map (black section) is represented by array [0,1,0,2,1,0,1,3,2,1,2,1]. In this case, 6 units of rain water (blue section) are being trapped.

Example 2:

Input: height = [4,2,0,3,2,5]

Output: 9

3)Valid Parenthesis

Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

An input string is valid if:

Open brackets must be closed by the same type of brackets.

Open brackets must be closed in the correct order.

Every close bracket has a corresponding open bracket of the same type.

Example 1:

Input: s = "()"

Output: true

Example 2:

Input: s = "()[]{}"

Output: true

Example 3:

Input: s = "(]"

Output: false

4)Sort Vowels in a string

Given a 0-indexed string s, permute s to get a new string t such that:

All consonants remain in their original places. More formally, if there is an index i with 0 <= i < s.length such that s[i] is a consonant, then t[i] = s[i].

The vowels must be sorted in the nondecreasing order of their ASCII values. More formally, for pairs of indices i, j with 0 <= i < j < s.length such that s[i] and s[j] are vowels, then t[i] must not have a higher ASCII value than t[j].

Return the resulting string.

The vowels are 'a', 'e', 'i', 'o', and 'u', and they can appear in lowercase or uppercase. Consonants comprise all letters that are not vowels.

Example 1:

Input: s = "lEetcOde"

Output: "lEOtcede"

Explanation: 'E', 'O', and 'e' are the vowels in s; 'l', 't', 'c', and 'd' are all consonants. The vowels are sorted according to their ASCII values, and the consonants remain in the same places.

Example 2:

Input: s = "lYmpH"

Output: "lYmpH"

Explanation: There are no vowels in s (all characters in s are consonants), so we return "lYmpH".

5)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"

6) Add Binary

Given two binary strings a and b, return their sum as a binary string.

Example 1:

Input: a = "11", b = "1"

Output: "100"

Example 2:

Input: a = "1010", b = "1011"

Output: "10101"

Constraints:

1 <= a.length, b.length <= 104

a and b consist only of '0' or '1' characters.

Each string does not contain leading zeros except for the zero itself.

7) Single Number

Given a non-empty array of integers nums, every element appears twice except for one. Find that single one. You must implement a solution with a linear runtime complexity and use only constant extra space.

Example 1:

Input: nums = [2,2,1]

Output: 1

Example 2:

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

Output: 4

Example 3:

Input: nums = [1]

Output: 1

Constraints:

1 <= nums.length <= 3 * 104

-3 * 104 <= nums[i] <= 3 * 104

8) find first and last position of element in sorted array

Given an array of integers nums sorted in non-decreasing order, find the starting and ending position of a given target value.

If target is not found in the array, return [-1, -1].

You must write an algorithm with **O(log n) runtime complexity.**

Example 1:

Input: nums = [5,7,7,8,8,10], target = 8

Output: [3,4]

Example 2:

Input: nums = [5,7,7,8,8,10], target = 6

Output: [-1,-1]

Example 3:

Input: nums = [], target = 0

Output: [-1,-1]

Constraints:

0 <= nums.length <= 105

-109 <= nums[i] <= 109

nums is a non-decreasing array.

-109 <= target <= 109

9)Maximun Subarray

Given an integer array nums, find the subarray with the largest sum, and return its sum.

Example 1:

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

Output: 6

Explanation: The subarray [4,-1,2,1] has the largest sum 6.

Example 2:

Input: nums = [1]

Output: 1

Explanation: The subarray [1] has the largest sum 1.

Example 3:

Input: nums = [5,4,-1,7,8]

Output: 23

Explanation: The subarray [5,4,-1,7,8] has the largest sum 23.

Constraints:

1 <= nums.length <= 105

-104 <= nums[i] <= 104

10) Print an Alternating Odd-Even Number Pattern

Given an Integer n.

Example 1:

Input: n= 6

Output:

1

2 4

1 3 5

2 4 6 8

1 3 5 7 9

2 4 6 8 10 12

11)Print Pyramid

Given an Integer n.

Example 1:

Input: n= 5

Output:

```
    *
   ***
  *****
 *******
*********
```

12)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"]

13)Two sum

Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to target.

You may assume that each input would have exactly one solution, and you may not use the same element twice.

You can return the answer in any order.

Example 1:

Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].

Example 2:
Input: nums = [3,2,4], target = 6
Output: [1,2]

Example 3:
Input: nums = [3,3], target = 6
Output: [0,1]

Constraints:
2 <= nums.length <= 104
-109 <= nums[i] <= 109
-109 <= target <= 109
Only one valid answer exists.

14)Best time to buy and sell stock

You are given an array prices where prices[i] is the price of a given stock on the ith day.

You want to maximize your profit by choosing a single day to buy one stock and choosing a different day in the future to sell that stock.

Return the maximum profit you can achieve from this transaction. If you cannot achieve any profit, return 0.

Example 1:

Input: prices = [7,1,5,3,6,4]

Output: 5

Explanation: Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6-1 = 5.

Note that buying on day 2 and selling on day 1 is not allowed because you must buy before you sell.

Example 2:

Input: prices = [7,6,4,3,1]

Output: 0

Explanation: In this case, no transactions are done and the max profit = 0.

15)First Missing Positive

Given an unsorted integer array nums. Return the smallest positive integer that is not present in nums.

You must implement an algorithm that runs in O(n) time and uses O(1) auxiliary space.

Example 1:

Input: nums = [1,2,0]

Output: 3

Explanation: The numbers in the range [1,2] are all in the array.

Example 2:

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

Output: 2

Explanation: 1 is in the array but 2 is missing.

Example 3:

Input: nums = [7,8,9,11,12]

Output: 1

Explanation: The smallest positive integer 1 is missing.

Constraints:

1 <= nums.length <= 105

-231 <= nums[i] <= 231 – 1

16) Count String

Write a program that takes a string as input and calculates the following:

The number of uppercase letters.

The number of lowercase letters.

The number of vowels.

The number of consonants

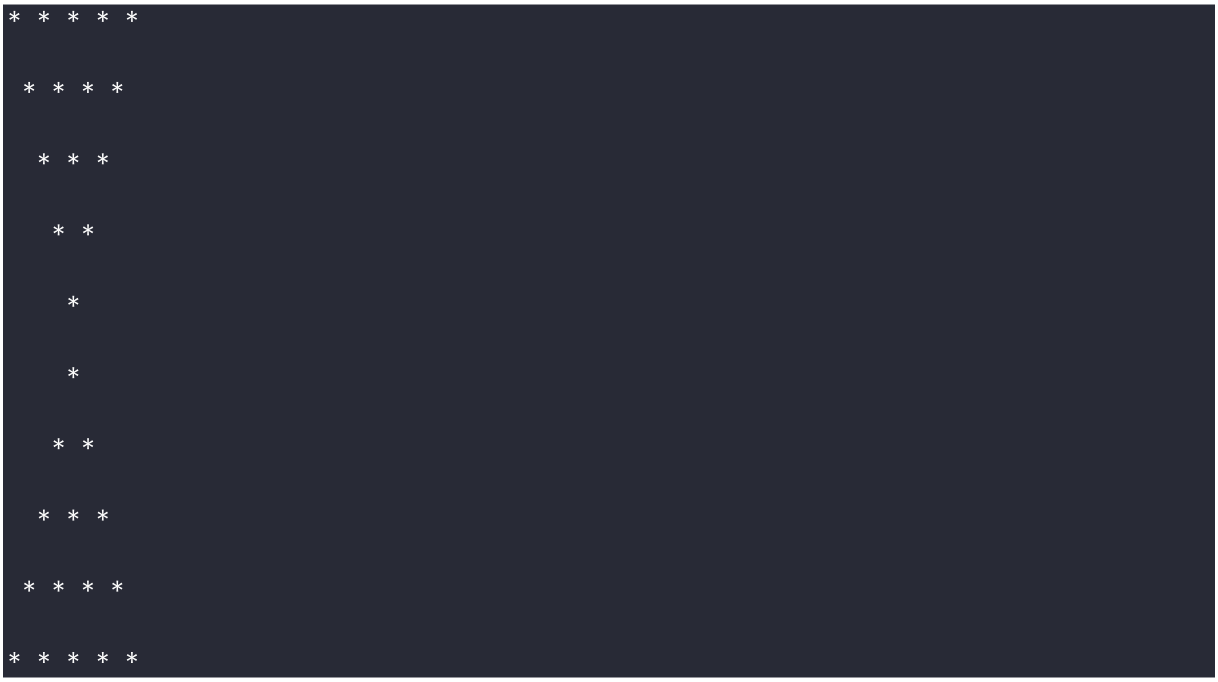
Example:

Sample input: Hello World

- Uppercase letters: 'H' and 'W' (total 2)

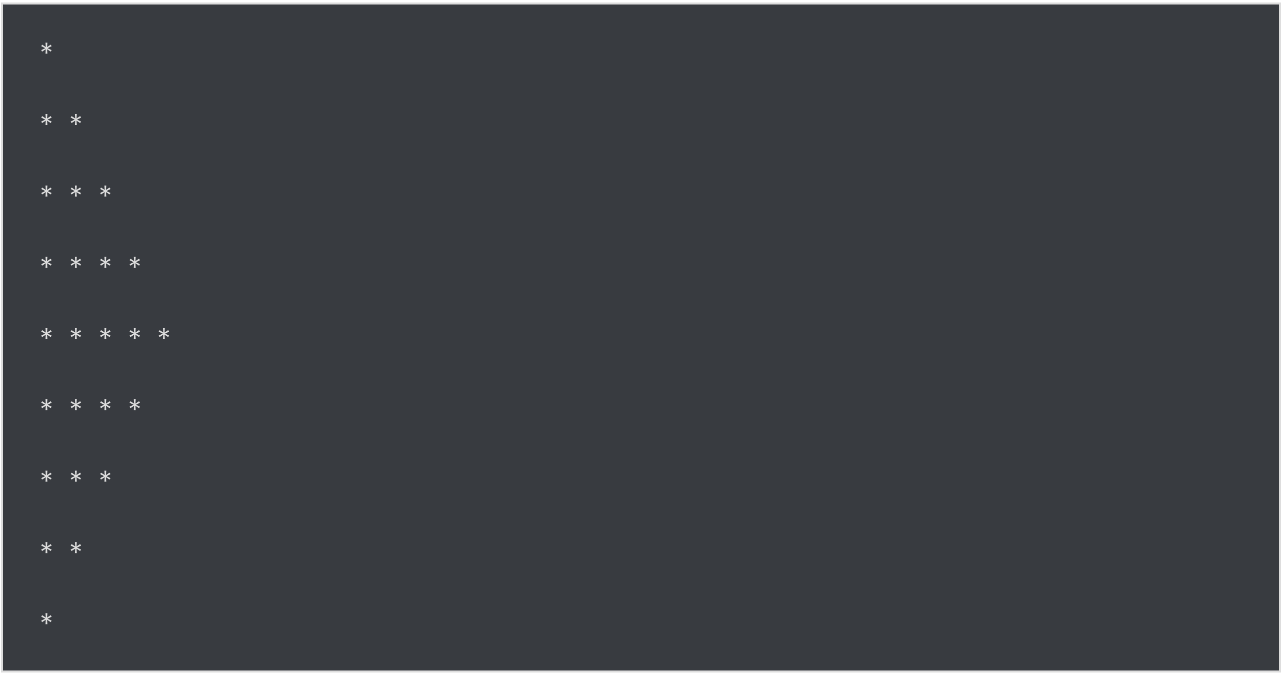
- Lowercase letters: 'e', 'l', 'l', 'o', 'o', 'r', 'l', 'd' (total 8)
- Vowels: 'e', 'o', 'o' (total 3)
- Consonants: 'H', 'l', 'l', 'W', 'r', 'l', 'd' (total 7)

17) Sandglass pattern of star



```
* * * * *
 * * * *
  * * *
   * *
    *
    *
   * *
  * * *
 * * * *
* * * * *
```

18) Right start pattern of star



```
*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*
```

19) Nim Game

You are playing the following Nim Game with your friend:

- Initially, there is a heap of stones on the table.
- You and your friend will alternate taking turns, and you go first.
- On each turn, the person whose turn it is will remove 1 to 3 stones from the heap.
- The one who removes the last stone is the winner.

Given n, the number of stones in the heap, return true if you can win the game assuming both you and your friend play optimally, otherwise return false.

Example 1:

Input: n = 4

Output: false

Explanation: These are the possible outcomes:

1. You remove 1 stone. Your friend removes 3 stones, including the last stone. Your friend wins.

2. You remove 2 stones. Your friend removes 2 stones, including the last stone. Your friend wins.

3. You remove 3 stones. Your friend removes the last stone. Your friend wins.

In all outcomes, your friend wins.

Example 2:

Input: n = 1

Output: true

Example 3:

Input: n = 2

Output: true

Constraints:

1 <= n <= 231 - 1

20)Bubble Sort Adhoc

Implement Bubble Sort and print the total number of swaps involved to sort the array.

Input Format

First line of input contains T - number of test cases. Its followed by 2T lines. First line of each test case contains N - size of the array. The next line contains N integers - elements of the array.

Constraints

1 <= T <= 100
1 <= N <= 100
-1000 <= ar[i] <= 1000

Output Format

For each test case, print the total number of swaps, separated by new line.

Sample Input 0

```
4
8
176 -272 -272 -45 269 -327 -945 176
2
-274 161
7
274 204 -161 481 -606 -767 -351
2
154 -109
```

Sample Output 0

```
15
0
16
1
```