

Infytq\_coding\_questions

# Question 1: Biggest Meatball

- **Problem Statement** – Bhojon is a restaurant company and has started a new wing in a city. They have every type of cook except the meatball artist. They had fired their last cook because the sale of meatballs in their restaurant is really great, and they can't afford to make meatballs again and again every time their stock gets empty. They have arranged a hiring program, where you can apply with their meatball.  
They will add the meatball in their seekh (a queue) and everytime they cut the meatball they take it and cut it on the day's quantity and then re-add the meatball in the seekh. You are the hiring manager there and you are going to say who is gonna be hired.
- Day's quantity means, on that very day the company sells only that kg of meatballs to every packet.
- If someone has less than a day's quantity, it will be counted as a sell.

Name	Type	Description
N	Integer	How many people are participating in the hiring process.
D	Integer	Day's quantity, how many grams of meatball is being sold to every packet.
Array[ ]	Integer array	Array of integers, the weight of meatballs everyone came with.

**Return:**

- The ith person whose meat is served at last

**Input Format:**

First line contains N.

Second line contains D.

After that N lines contain The ith person's meatball weight.

**Output Format:** The 1 based index of the man whose meatball is served at the last.

**Sample Input 1:**

4

2

[7 8 9 3]

**Sample Output 1:**

3

**Explanation:**

The seekh or meatball queue has [7 8 9 3] this distribution. At the first serving they will cut 2 kgs of meatball from the first meatball and add it to the last of the seekh, so after 1st time it is:

[8 9 3 5]

Then, it is: [9 3 5 6], [3 5 6 7], [5 6 7 1], [6 7 1 3], [7 1 3 4], [1 3 4 5], [3 4 5], [4 5 1], [5 1 2], [1 2 3], [2 3], [3], [1], [0]

So the last served meatball belongs to the 3rd person.

### **Constraints:**

$$1 \leq N \leq 10^3$$

$$1 \leq D \leq 10^3$$

$$1 \leq \text{Array}[i] \leq 10^3$$





# Self Sufficient

- **Problem Statement** – Abhijeet is one of those students who tries to get his own money by part time jobs in various places to fill up the expenses for buying books. He is not placed in one place, so what he does, he tries to allocate how much the book he needs will cost, and then work to earn that much money only. He works and then buys the book respectively. Sometimes he gets more money than he needs so the money is saved for the next book. Sometimes he doesn't. In that time, if he has stored money from previous books, he can afford it, otherwise he needs money from his parents.
- Now His parents go to work and he can't contact them amid a day. You are his friend, and you have to find how much money minimum he can borrow from his parents so that he can buy all the books.
- He can Buy the book in any order.



Name	Type	Description
N	Integer	How many Books he has to buy that day.
EarnArray[ ]	Integer array	Array of his earnings for the ith book
CostArray[ ]	Integer array	Array of the actual cost of the ith book.

**Output Format:** The minimum money he needs to cover the total expense.

**Sample Input 1:**

3

[3 4 2]

[5 3 4]

**Sample Output 1:**

3

## Constraints:

$$1 \leq N \leq 10^3$$

$$1 \leq \text{EarnArray}[i] \leq 10^3$$

$$1 \leq \text{CostArray}[i] \leq 10^3$$



# Concatenation

Input: a string of comma separated numbers. The numbers 5 and 8 are present in the list Assume that 8 always comes after 5.

Case 1: num1 = add all numbers which do not lie between 5 and 8 in the input.

Case 2: num2= numbers formed by concatenating all numbers from 5 to 8.

Output: sum of num1 and num2

Example: 1) 3,2,6,5,1,4,8,9

Num1 :  $3+2+6+9=20$

Num2: 5148

output:  $5248+20=5168$

**Answer: 13**



# Unique Elements

Take input a number 'N' and an array as given below.

Input:-N = 2

Array =1,2,3,3,4,4

O/p : 2

Find the least number of unique elements after deleting N numbers of elements from the array.

In the above example, after deleting N=2 elements from the array.

In above 1,2 will be deleted.

So 3,3,4,4 will be remaining so,

2 unique elements are in the array i.e 3 and 4.





# Largest Even Number

A string which is a mixture of letters, numbers, and special characters from which produce the largest

even number from the available digit after removing the duplicates digits.

If an even number did not produce then return -1.

Ex: infosys@337

O/p : -1

.....

Hello#81@21349

O/p : 984312



## InfyTQ 2019 : Find the position from where the parenthesis is not balanced

Given a string **str** consisting of parenthesis from [ "(", ")", "{", "}", "[", "]" ].

If the **String** is perfectly balanced return 0 else return the index(starting from 1)at which the nesting is found to be wrong.

**Examples:**

**Input :** str = "{[()]}"

**Output :** 0

**Input :** str = "{[]()}"

**Output :** 3

**Input :** str = "}[]{}"

**Output :** 1

**Input :** str = "{([)]{"

**Output :** 7



# OTP Generation

You will be given a number in the form of string, extract out digits at odd places, square & merge them. First 4 digits will be the required OTP.

Input :

First Input : String

Output : 4 digit OTP

Sample Testcases :

I/P 1:

345679

O/P 1:

1636



# Program to check for Pronic Number

Write a Program in Java to input a number and check whether it is a **Pronic Number** or **Heteromecic Number** or not.

**Pronic Number** : A pronic number, oblong number, rectangular number or heteromecic number, is a number which is the product of two consecutive integers, that is,  $n(n + 1)$ .

The first few pronic numbers are:

0, 2, 6, 12, 20, 30, 42, 56, 72, 90, 110, 132, 156, 182, 210, 240, 272, 306, 342, 380, 420, 462 ... etc.





# Largest subset whose all elements are Fibonacci numbers

Given an array with positive number the task is that we find largest subset from array that contain elements which are Fibonacci numbers.

Input : `arr[] = {1, 4, 3, 9, 10, 13, 7};`

Output : `subset[] = {1, 3, 13}`

The output three numbers are Fibonacci numbers.

Input : `arr[] = {0, 2, 8, 5, 2, 1, 4, 13, 23};`

Output : `subset[] = {0, 2, 8, 5, 2, 1, 13, 23}`



# Even Odd

- Given a string of special characters, digits and alphabets. If number of special characters are odd, output a list containing sequence of odd and even numbers. If special characters are even, output a list containing sequence of even and odd numbers. If there are remaining numbers then print them at last.
- @@#\$23345w6w6
- Output: 2343656



# Longest prefix which is also suffix

- Given a string s, find the length of the longest prefix, which is also a suffix. The prefix and suffix should not overlap.

Input : aabcdaabc

Output : 4

The string "aabc" is the longest prefix which is also suffix.

Input : abcab

Output : 2

Input : aaaa

Output : 2



# Find maximum (or minimum) sum of a subarray of size k

- Given an array of integers and a number k, find the maximum sum of a subarray of size k

Input : arr[] = {100, 200, 300, 400}

k = 2

Output : 700

Input : arr[] = {1, 4, 2, 10, 23, 3, 1, 0, 20}

k = 4

Output : 39

We get maximum sum by adding subarray {4, 2, 10, 23} of size 4.

Input : arr[] = {2, 3}

k = 3

Output : Invalid

There is no subarray of size 3 as size of whole array is 2.





# Zig Zag array

Given an array **Arr** (distinct elements) of size **N**. Rearrange the elements of array in zig-zag fashion. The converted array should be in form **a < b > c < d > e < f**. The relative order of elements is same in the output i.e you have to iterate on the original array only.

**Example 1:**

**Input:** N = 7 Arr[] = {4, 3, 7, 8, 6, 2, 1} **Output:** 3 7 4 8 2 6 1

**Example 2:**

**Input:** N = 4 Arr[] = {1, 4, 3, 2} **Output:** 1 4 2 3

**Your Task:**

You don't need to read input or print anything. Your task is to complete the function **zigZag()** which takes the array of integers **arr** and **n** as parameters and returns void. You need to modify the array itself.

**Expected Time Complexity:** O(N)

**Expected Auxiliary Space:** O(1)

**Constraints:**

$1 \leq N \leq 10^5$

$0 \leq \text{Arr}_i \leq 10^6$



Given an array of integers and a number K. Find the count of distinct elements in every window of size K in the array.

**Example 1:**

**Input:** N = 7, K = 4 A[] = {1,2,1,3,4,2,3}

**Output:** 3 4 4 3

**Example 2:**

**Input:** N = 3, K = 2 A[] = {4,1,1}

**Output:** 2 1

**Your Task:**

Your task is to complete the function **countDistinct()** which takes the array A[], the size of the array(N) and the window size(K) as inputs and returns an array containing the count of distinct elements in every contiguous window of size K in the array A[].

**Constraints:**

$1 \leq K \leq N \leq 10^5$

$1 \leq A[i] \leq 10^5$ , for each valid i



# Pattern Matching

- Given a text  $txt[0..n-1]$  and a pattern  $pat[0..m-1]$ , write a function  $search(char\ pat[], char\ txt[])$  that prints all occurrences of  $pat[]$  in  $txt[]$ . You may assume that  $n > m$ .

Input:

```
txt[] = "THIS IS A TEST TEXT"
```

```
pat[] = "TEST"
```

Output: Pattern found at index 10

Input:

```
txt[] = "AABAACAADAABAABA"
```

```
pat[] = "AABA"
```

Output: Pattern found at index 0 Pattern found at index 9 Pattern found at index 12



# Six Prime

- Given a range, find the prime pair whose difference is six
- For example:
- 1-14
- 5 and 11, 7 and 13





# Stepping Numbers

Given two integers 'n' and 'm', find all the stepping numbers in range [n, m]. A number is called **stepping number** if all adjacent digits have an absolute difference of 1. 321 is a Stepping Number while 421 is not.

**Input** : n = 0, m = 21

**Output** : 0 1 2 3 4 5 6 7 8 9 10 12 21

**Input** : n = 10, m = 15

**Output** : 10, 12



# Remove Invalid Parentheses

- An expression will be given which can contain open and close parentheses and optionally some characters, No other operator will be there in string. We need to remove minimum number of parentheses to make the input string valid. If more than one valid output are possible removing same number of parentheses then print all such output.

Input : str = "()())()"

Output : ()()() (())()

Input : str = (v)())()

Output : (v)()() (v())()



# Minimum Gifts

A Company has decided to give some gifts to all of its employees. For that, the company has given some rank to each employee. Based on that rank, the company has made certain rules to distribute the gifts.

The rules for distributing the gifts are:

Each employee must receive at least one gift.

Employees having higher ranking get a greater number of gifts than their neighbours.

What is the minimum number of gifts required by the company?

## **Constraints**

$$1 < T < 10$$

$$1 < N < 100000$$

$$1 < \text{Rank} < 10^9$$

## **Input**

First line contains integer T, denoting the number of test cases.

For each test case:

First line contains integer N, denoting the number of employees.

Second line contains N space separated integers, denoting the rank of each employee.

## **Output**

For each test case print the number of minimum gifts required on a new line.

### **Example 1**

#### **Input**

```
2
5
1 2 1 5 2
2
1 2
```

#### **Output**

```
7
3
```



- Roco is an island near Africa which is very prone to forest fire. Forest fire is such that it destroys the complete forest. Not a single tree is left. This island has been cursed by God, and the curse is that whenever a tree catches fire, it passes the fire to all its adjacent trees in all 8 directions, North, South, East, West, North-East, North-West, South-East, and South-West. And it is given that the fire is spreading every minute in the given manner, i.e. every tree is passing fire to its adjacent trees. Suppose that the forest layout is as follows where T denotes tree and W denotes water.
- Your task is that given the location of the first tree that catches fire, determine how long it would take for the entire forest to be on fire. You may assume that the layout of the forest is such that the whole forest will catch fire for sure and that there will be at least one tree in the forest.



### **Input Format:**

First line contains two integers, M, N, space separated, giving the size of the forest in terms of the number of rows and columns respectively.

The next line contains two integers X,Y, space separated, giving the coordinates of the first tree that catches the fire.

The next M lines, where ith line containing N characters each of which is either T or W, giving the position of the Tree and Water in the ith row of the forest.

### **Output Format:**

Single integer indicating the number of minutes taken for the entire forest to catch fire

### **Constrains:**

$$3 \leq M \leq 20$$

$$3 \leq N \leq 20$$

Sample Input 1:

3 3

W T T

T W W

W T T

Sample Output 1:

5

Sample Input 2:

6 6

1 6

W T T T T T

T W W W W W

W T T T T T

W W W W W T

T T T T T T

T W W W W W

Sample Output 2:

16



**Problem Description -:** In this 3 Palindrome, Given an input string word, split the string into exactly 3 palindromic substrings. Working from left to right, choose the smallest split for the first substring that still allows the remaining word to be split into 2 palindromes.

Similarly, choose the smallest second palindromic substring that leaves a third palindromic substring.

If there is no way to split the word into exactly three palindromic substrings, print “Impossible” (without quotes). Every character of the string needs to be consumed.

Cases not allowed –

After finding 3 palindromes using above instructions, if any character of the original string remains unconsumed.

No character may be shared in forming 3 palindromes.

## **Constraints**

1 <= the length of input string <= 1000

## **Input**

First line contains the input string consisting of characters between [a-z].

## **Output**

Print 3 substrings one on each line.

## **Example 1**

### **Input**

nayannamantenet

### **Output**

nayan

naman

tenet