Ex. No. : 5.1 **Date:**

Register No.: **Name:**

Balanced Array

Given an array of numbers, find the index of the smallest array element (the pivot), for which the sums of all elements to the left and to the right are equal. The array may not be reordered.

Example

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

- the sum of the first three elements, 1+2+3=6. The value of the last element is 6.
- Using zero based indexing, arr[3]=4 is the pivot between the two subarrays.
- The index of the pivot is 3.

Constraints

- $3 \leq n \leq 10^5$
- $1 \leq arr[i] \leq 2 \times 10^4$, where $0 \leq i < n$
- It is guaranteed that a solution always exists.

The first line contains an integer n, the size of the array arr.

Each of the next n lines contains an integer, arr[i], where $0 \leq i < n$.

Ex. No. : 5.2 **Date:**

Register No.: **Name:**

Check pair with difference k

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that A[i] - A[j] = k, i != j.

Input Format

1. First line is number of test cases T. Following T lines contain:
2. N, followed by N integers of the array
3. The non-negative integer k

Output format

Print 1 if such a pair exists and 0 if it doesn't.

```
test=int(input())
while(test>0):
    tot=int(input())
    a=[]
    count=0
    for i in range(tot):
        a.append(int(input()))
    k=int(input())
    for i in range(tot):
        for j in range(i+1,tot):
            if(abs(a[i]-a[j])==k):
                count=count+1
                break
    if(count!=0):
        print(1)
    else:
        print(0)
    test=test-1
```

Ex. No. : 5.3 **Date:**

Register No.: **Name:**

Count Elements

Complete the program to count frequency of each element of an array. Frequency of a particular element will be printed once.

```
r = int(input())
a = []
b=[]

for i in range(r):
    a.append(int(input()))

b=a.copy()

b2 = []
for i in b:
    if i not in b2:
        b2.append(i)


for i in range(len(b2)):
    count=0
    for j in range(r):
        if(b2[i]==a[j]):
            count=count+1
    print(b2[i],"occurs",count,"times")
```

Ex. No. : 5.4 **Date:**

Register No.: **Name:**

Distinct Elements in an Array

Program to print all the distinct elements in an array. Distinct elements are nothing but the unique (non-duplicate) elements present in the given array.

Input Format:

First line take an Integer input from stdin which is array length n.

Second line take n Integers which is inputs of array.

Output Format:

Print the Distinct Elements in Array in single line which is space Separated

```
n=int(input())
a=list()

for i in range(n):
    a.append(int(input()))

for i in range(n):
    for j in range(i+1,n):
        x=a[i]
        if(x==a[j]):
            a[j]=0

for i in range(n):
    if(a[i]!=0):
        print(a[i],end=' ')
```

Ex. No. : 5.5 **Date:**

Register No.: **Name:**

Element Insertion

Consider a program to insert an element / item in the sorted array. Complete the logic by filling up required code in editable section. Consider an array of size 10. The eleventh item is the data is to be inserted.

```
def insert_in_sorted_array(arr, item):
    position = 0
    while position < len(arr) and arr[position] < item:
        position += 1
    arr.insert(position, item)
    return arr

arr = []
for _ in range(10):
    arr.append(int(input()))

item = int(input("Enter the item to be inserted: "))
print(f"ITEM to be inserted: {item}")

new_arr = insert_in_sorted_array(arr, item)

print("After insertion array is:")
for num in new_arr:
    print(num)
```

Ex. No. : **5.6** **Date:**

Register No.: **Name:**

Find the Factor

Determine the factors of a number (i.e., all positive integer values that evenly divide into a number) and then return the p^{th} element of the list, sorted ascending. If there is no p^{th} element, return 0.

Constraints

$1 \leq n \leq 10^{15}$

$1 \leq p \leq 10^{9}$

The first line contains an integer n, the number to factor.

The second line contains an integer p, the 1-based index of the factor to return.

```
n=int(input())
m=int(input())
a=[]

for i in range(1,n+1):
    if(n%i==0):
        a.append(i)

m=m-1
length=len(a)

if(m < len(a)):
    print(a[m])
else:
    print('0')
```

Ex. No. : 5.7 Date:

Register No.: Name:

Merge List

Write a Python program to Zip two given lists of lists.

Input:
m : row size
n: column size
list1 and list 2 : Two lists
Output
Zipped List : List which combined both list1 and list2

```
m=int(input())
n=int(input())
a=[]
b=[]
m1=m

for _ in range(m):
    r=[]
    for _ in range(n):
        r.append(int(input()))
    if(m==m1):
        a.append(r)
    else:
        a=a+r

for _ in range(m):
    r=[]
    for _ in range(n):
        r.append(int(input()))
    if(m==m1):
        b.append(r)
    else:
        b=b+r

for i in range(n):
    a[i].extend(b[i])
```

Ex. No. : 5.8 Date:

Register No.: Name:

Merge Two Sorted Arrays Without Duplication

Output is a merged array without duplicates.

Input Format

N1 - no of elements in array 1

Array elements for array 1

N2 - no of elements in array 2

Array elements for array2

Output Format

Display the merged array

```
N1 = int(input())
array1 = []
for _ in range(N1):
    array1.append(int(input()))

N2 = int(input( ))
array2 = []
for _ in range(N2):
    array2.append(int(input()))

merged_array = array1 + array2

merged_array = list(set(merged_array))

merged_array.sort()

print(merged_array)
```

Ex. No. : 5.9 **Date:**

Register No.: **Name:**

Print Element Location

Write a program to print all the locations at which a particular element (taken as input) is found in a list and also print the total number of times it occurs in the list. The location starts from 1.

```
def find_element_locations(arr, target):
    locations = [i + 1 for i, x in enumerate(arr) if x == target]
    count = len(locations)
    return count, locations

n = int(input())

arr = []
for _ in range(n):
    arr.append(int(input()))

target = int(input())

count, locations = find_element_locations(arr, target)

if count > 0:
    for loc in locations:
        print(f"{target} is present at location {loc}.")
    print(f"{target} is present {count} times in the array.")
else:
    print(f"{target} is not present in the array.")
```

Ex. No. : 5.10 **Date:**

Register No.: **Name:**

Strictly increasing

Write a Python program to check if a given list is strictly increasing or not. Moreover, If removing only one element from the list results in a strictly increasing list, we still consider the list true

Input:
n : Number of elements
List1: List of values
Output
Print "True" if list is strictly increasing or decreasing else print "False"

```
n = int(input())
l1 = [int(input()) for _ in range(n)]
is_incre=True

for i in range(1, n):
  if l1[i]<=l1[i-1]:
    is_incre = False
    break

if is_incre:
  print("True")
elif l1 == sorted(l1, reverse=True):
  print("True")
else:
  for i in range(n):
    temp = l1[:i] + l1[i+1:]
    if temp == sorted (temp):
      print("True")
    break
  else:
    print("False")
```