# [05 - List in Python](https://www.rajalakshmicolleges.net/moodle/course/view.php?id=84&section-5)

Sample Case 0

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

90

Sample Input 0

4

1

2

3

3

Sample Output 0

2

Explanation 0

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

Sample Case 1

Sample Input 1

3

1

2

1

Sample Output 1

1

Explanation 1

* The first and last elements are equal to 1.
* Using zero based indexing, arr[1]=2 is the pivot between the two subarrays.
* The index of the pivot is 1.

##### For example:

|  |  |
| --- | --- |
| **Input** | **Result** |
| 4  1  2  3  3 | 2 |
| 3  1  2  1 | 1 |

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

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

91

**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 ≤ n ≤ 105
* 1 ≤ arr[i] ≤ 2 × 104, where 0 ≤ 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 ≤ i < n.

n = int(input())

arr = []

for i in range(n):

arr.append(int(input()))

total = sum(arr)

left = 0

for i in range(n):

total -= arr[i]

if left == total:

print(i)

break

left += arr[i]

Input 1

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

92

3

1

3

5

4

Output:

1

Input 1

3

1

3

5

99

Output 0

##### For example:

|  |  |
| --- | --- |
| **Input** | **Result** |
| 1  3  1  3  5  4 | 1 |
| 1  3  1  3  5  99 | 0 |

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

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

93

**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.

t=int(input())

while t:

t-=1

n=int(input())

nums=[int(input()) for i in range(n)]

k=int(input())

flag=False

for i in nums:

comp=abs(i-k)

if comp in nums:

flag=True

break

if flag:

print(1)

else:

print(0)

Sample Test Cases Test Case 1

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

94

Input 7

23

45

23

56

45

23

40

Output

23 occurs 3 times

45 occurs 2 times

56 occurs 1 times

40 occurs 1 times

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

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

95

**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.

n = int(input())

arr = [int(input()) for \_ in range(n)]

# Counting frequency

frequency = {}

for num in arr:

if num in frequency:

frequency[num] += 1

else:

frequency[num] = 1

# Output

for num, freq in frequency.items():

print(f"{num} occurs {freq} times")

Example Input: 5

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

96

1

2

2

3

4

Output:

1 2 3 4

Example Input: 6

1

1

2

2

3

3

Output:

1 2 3

For example: Input Result 5

1

2

2

3

4

1 2 3 4

6

1

1

2

2

3

3

1 2 3

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

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

97

**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

t=int(input())

while t:

t-=1

n=int(input())

nums=[int(input()) for i in range(n)]

k=int(input())

flag=False

for i in nums:

comp=abs(i-k)

if comp in nums:

flag=True

break

if flag:

print(1)

else:

print(0)

Sample Test Cases Test Case 1

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

98

Input 1

3

4

5

6

7

8

9

10

11

2

Output

ITEM to be inserted:2 After insertion array is: 1

2

3

4

5

6

7

8

9

10

11

Test Case 2 Input

11

22

33

55

66

77

88

99

110

120

44

Output

ITEM to be inserted:44 After insertion array is: 11

22

33

44

55

66

77

88

99

110

120

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

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

99

**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.

# Initialize a sorted array of size 10

arr = []

# Read the initial sorted array of size 10

for \_ in range(10):

arr.append(int(input()))

# Read the element to be inserted

item\_to\_insert = int(input())

# Insertion logic

inserted = False

for i in range(10):

if arr[i] > item\_to\_insert:

# Shift elements to the right to make space for insertion

arr.insert(i, item\_to\_insert)

inserted = True

break

# If the new element is greater than all existing elements, insert at the end

if not inserted:

arr.append(item\_to\_insert)

# Print the updated array after insertion

print(f"ITEM to be inserted:{item\_to\_insert}")

print("After insertion array is:")

for num in arr:

print(num)

##### Sample Case 0

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

100

**Sample Input 0**

10

3

##### Sample Output 0

5

##### Explanation 0

Factoring n = 10 results in {1, 2, 5, 10}. Return the p = 3rd factor, 5, as the answer.

##### Sample Case 1

**Sample Input 1**

10

5

##### Sample Output 1

0

##### Explanation 1

Factoring n = 10 results in {1, 2, 5, 10}. There are only 4 factors and p = 5, therefore 0 is returned as the answer.

##### Sample Case 2

**Sample Input 2**

1

1

##### Sample Output 2

1

##### Explanation 2

Factoring n = 1 results in {1}. The p = 1st factor of 1 is returned as the answer.

##### For example:

|  |  |
| --- | --- |
| **Input** | **Result** |
| 10  3 | 5 |
| 10  5 | 0 |
| 1  1 | 1 |

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

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

101

**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 pth element of the [list](http://118.185.187.137/moodle/mod/resource/view.php?id=732), sorted ascending. If there is no pth element, return 0.

##### Constraints

1 ≤ n ≤ 1015

1 ≤ p ≤ 109

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.

a=int(input())

b=int(input())

li=[]

i=1

while i<=a:

if a%i==0:

li.append(i)

i+=1

if b<=len(li):

print(li[b-1])

else:

print(0)

Sample test case

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

102

Sample input 2

2

1

3

5

7

2

4

6

8

Sample Output

[[1, 3, 2, 4], [5, 7, 6, 8]]

**Ex. No. : 5.7 Date:**

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

103

**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())

list1 = []

list2 = []

for i in range(m):

row = []

for j in range(n):

row.append(int(input()))

list1.append(row)

for i in range(m):

row = []

for j in range(n):

row.append(int(input()))

list2.append(row)

zipp = []

for row1, row2 in zip(list1, list2):

zipp.append(row1 + row2)

print(zipp)

Sample Input 1

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

104

5

1

2

3

6

9

4

2

4

5

10

Sample Output 1

1 2 3 4 5 6 9 10

**Ex. No. : 5.8 Date:**

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

105

**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

a=int(input())

s1=[]

for i in range(a):

b=int(input())

s1.append(b)

a=int(input())

for i in range(a):

b=int(input())

if b not in s1:

s1.append(b)

s1.sort()

for i in s1:

print(i,end=' ')

For example, if there are 4 elements in the array: 5

6

5

7

If the element to search is 5 then the output will be: 5 is present at location 1

5 is present at location 3

5 is present 2 times in the array. Sample Test Cases

Test Case 1 Input

4

5

6

5

7

5

Output

5 is present at location 1.

5 is present at location 3.

5 is present 2 times in the array.

Test Case 2 Input

5

67

80

45

97

100

50

Output

50 is not present in the array.

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

106

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

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

107

**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.

# Initialize a sorted array of size 10

arr = []

# Read the initial sorted array of size 10

for \_ in range(10):

arr.append(int(input()))

# Read the element to be inserted

item\_to\_insert = int(input())

# Insertion logic

inserted = False

for i in range(10):

if arr[i] > item\_to\_insert:

# Shift elements to the right to make space for insertion

arr.insert(i, item\_to\_insert)

inserted = True

break

# If the new element is greater than all existing elements, insert at the end

if not inserted:

arr.append(item\_to\_insert)

# Print the updated array after insertion

print(f"ITEM to be inserted:{item\_to\_insert}")

print("After insertion array is:")

for num in arr:

print(num)

Sample Test Case Input

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

108

7

1

2

3

0

4

5

6

Output True

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

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

109

**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())

lst=[]

flag=0

for i in range(n):

y=int(input())

lst.append(y)

if lst==sorted(lst) or lst==sorted(lst,reverse=True):

flag=1

for i in range(n):

dup=lst.copy()

dup.pop(i)

if dup==sorted(dup) or dup==sorted(dup,reverse=True):

flag=1

print(flag==1)

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

110