.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 96



05 - List in Python

Sample Case 0

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 97



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 | 2 |
| 1 |  |
| 2 |  |
| 3 |  |
| 3 |  |
| 3 | 1 |
| 1 |  |
| 2 |  |
| 1 |  |

**Ex. No. : 5.1 Date:16.04.2024**

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 98



**Register No.:231401009 Name:M.ANJALI**

# 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.

CODE:

def find\_pivot\_index(arr): total\_sum = sum(arr) left\_sum = 0

for i, num in enumerate(arr): total\_sum -= num

if left\_sum == total\_sum: return i

left\_sum += num

return -1 # If no such pivot exists

n = int(input())

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

.

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 99



pivot\_index = find\_pivot\_index(arr)

print(pivot\_index)

Input 1

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 100



3

1

3

5

4

Output:

1

Input 1

3

1

3

5

99

Output 0

## For example:

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

**Ex. No. : 5.2 Date: 16.04.2024**

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 101



**Register No.:231401009 Name:M.ANJALI**

# 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.

CODE :

def find\_pair\_with\_difference(arr, k): i, j = 0, 1

while j < len(arr): diff = arr[j] - arr[i] if diff == k:

return 1 elif diff < k:

j += 1

else:

i += 1

if i == j:

j += 1

return 0

.

# Input

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 102



T = int(input()) for \_ in range(T):

N = int(input())

array = [int(input()) for i in range(N)]

k = int(input())

# Output print(find\_pair\_with\_difference(array, k))

Sample Test Cases Test Case 1

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 103



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: 16.04.2024**

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 104



**Register No.:231401009 Name:M.ANJALI**

# Count Elements

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

CODE’:

def count\_frequencies(arr):

# Dictionary to store the frequency of each element frequency\_dict = {}

# Count the frequency of each element for element in arr:

if element in frequency\_dict: frequency\_dict[element] += 1

else:

frequency\_dict[element] = 1

# Print the frequency of each element once

for element, frequency in frequency\_dict.items(): print(f"Element {element} appears {frequency} times")

Example Input: 5

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 105



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: 16.04.2024**

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 106



**Register No.:231401009 Name:M.ANJALI**

# 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

CODE:

def print\_distinct\_elements(arr): distinct\_elements = list(set(arr)) for element in distinct\_elements:

print(element,end=" ")

# Test the function array = [] n=int(input())

for i in range(0,n): ele=int(input()) array.append(ele)

print\_distinct\_elements(array)

Sample Test Cases Test Case 1

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 107



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: 16.04.2024**

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 108



**Register No.:231401009 Name:M.ANJALI**

# 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.

CODE:

def insert\_into\_sorted\_array(arr, item):

if len(arr) >= 11: return arr

arr.append(item) arr.sort()

return arr

arr1= [int(input()) for i in range(10)] item1 = int(input())

print("ITEM to be inserted:",item1,sep="") print("After insertion array is:")

print(\*insert\_into\_sorted\_array(arr1, item1),sep="\n")

## Sample Case 0

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 109



**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: 16.04.2024**

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 110



**Register No.:231401009 Name:M.ANJALI**

# 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, 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.

CODE:

n = int(input()) p = int(input())

factors = [i for i in range(1, n + 1) if n % i == 0] factors.sort()

result = factors[p - 1] if p <= len(factors) else 0 print(result)

Sample test case

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 111



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: 16.04.2024**

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 112



**Register No.:231401009 Name:M.ANJALI**

# 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

CODE:

def zip\_lists(list1, list2): zipped\_list = []

for sublist1, sublist2 in zip(list1, list2): zipped\_list.append(sublist1 + sublist2)

return zipped\_list

def main():

m = int(input(""))

n = int(input("")) list1 = []

for \_ in range(m):

sublist = [int(input()) for \_ in range(n)] list1.append(sublist)

.

list2 = []

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 113



for \_ in range(m):

sublist = [int(input()) for \_ in range(n)] list2.append(sublist)

zipped\_list = zip\_lists(list1, list2) print(zipped\_list)

if name == " main ": main()

Sample Input 1

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 114



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: 16.04.2024**

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 115



**Register No.:231401009 Name:M.ANJALI**

# 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

CODE:

def merge\_arrays(arr1, arr2):

set1 = set(arr1) set2 = set(arr2)

merged\_array = sorted(set1.union(set2))

return merged\_array

def main(): try:

n1 = int(input())

.

arr1 = [int(input()) for \_ in range(n1)]

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 116



n2 = int(input())

arr2 = [int(input()) for \_ in range(n2)]

merged = merge\_arrays(arr1, arr2)

print(end="")

for num in merged: print(num, end=" ")

except ValueError: print()

if name == " main ": main()

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

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 117



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.

.

**Ex. No. : 5.9 Date: 16.04.2024**

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 118



**Register No.:231401009 Name:M.ANJALI**

# 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.

CODE:

def find\_element\_locations(arr, element):

locations = [i+1 for i, x in enumerate(arr) if x == element] count = len(locations)

if count > 0:

for loc in locations:

print(f"{element} is present at location {loc}.") print(f"{element} is present {count} times in the array.")

else:

print(f"{element} is not present in the array.")

n=int(input()) arr = []

for i in range(0,n): ele=int(input()) arr.append(ele)

element=int(input()) find\_element\_locations(arr, element)

Sample Test Case Input

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 119



7

1

2

3

0

4

5

6

Output True

**Ex. No. : 5.10 Date: 16.04.2024**

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 120



**Register No.:231401009 Name:M.ANJALI**

# 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"

CODE:

def is\_strictly\_increasing(lst): def is\_increasing(lst):

return all(x < y for x, y in zip(lst, lst[1:]))

# Check if the list is already strictly increasing if is\_increasing(lst):

return True

# Check if removing one element can make the list strictly increasing for i in range(len(lst)):

if is\_increasing(lst[:i] + lst[i+1:]): return True

return False

def is\_strictly\_decreasing(lst): def is\_decreasing(lst):

return all(x > y for x, y in zip(lst, lst[1:]))

# Check if the list is already strictly decreasing if is\_decreasing(lst):

return True

# Check if removing one element can make the list strictly decreasing for i in range(len(lst)):

if is\_decreasing(lst[:i] + lst[i+1:]): return True

.

return False

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 121



# Input

n = int(input())

List1 = [int(input()) for \_ in range(n)]

# Output

print("True" if is\_strictly\_increasing(List1) or is\_strictly\_decreasing(List1) else "False")

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College** 122

