# 

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

# 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 ≤ 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.

# Sample Case 0

# 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 |
| --- | --- |
| 41233 | 2 |
| 3121 | 1 |

# PROGRAM:

# 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

# 

# 

# n=int(input())

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

# print(find\_pivot\_index(arr))

# 

# 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.

# Input

# 1

# 3

# 1

# 3

# 5

# 4

# Output:

# 1

# Input

# 1

# 3

# 1

# 3

# 5

# 99

# Output

# 0

# 

# For example:

| Input | Result |
| --- | --- |
| 131354 | 1 |
| 1313599 | 0 |

# PROGRAM:

# t=int(input())

# for i in range(0,t):

# n=int(input())

# l=[]

# for j in range(0,n):

# a=int(input())

# l.append(a)

# p=int(input())

# for k in range(0,n):

# c=0

# for m in range(i+1,n):

# if l[m]-l[k]==p:

# c=1

# print('1')

# break

# if c==1:

# break

# if c==0:

# print('0')

# #print('\n')

# 

# Ex. No. : 5.3 Date:

# Register No.: Name:

Given two lists A and B, and B is an anagram of A. B is an anagram of A means B is made by randomizing the order of the elements in A.

We want to find an *index mapping* P, from A to B. A mapping P[i] = j means the ith element in A appears in B at index j.

These lists A and B may contain duplicates. If there are multiple answers, output any of them.

For example, given

**Input**

5

12 28 46 32 50

50 12 32 46 28

**Output**

1 4 3 2 0

**Explanation**

A = [12, 28, 46, 32, 50]

B = [50, 12, 32, 46, 28]

We should return

[1, 4, 3, 2, 0]

as P[0] = 1 because the 0th element of A appears at B[1], and P[1] = 4 because the 1st element of A appears at B[4], and so on.

**Note:**

1. A, B have equal lengths in range [1, 100].
2. A[i], B[i] are integers in range [0, 10^5].

# 

# PROGRAM:

# def index\_mapping(A,B):

# index\_map={num:i for i,num in enumerate(B)}

# return ' '.join(str(index\_map[num]) for num in A)

# 

# n=int(input())

# A=list(map(int, input().split()))

# B=list(map(int, input().split()))

# print(index\_mapping(A,B))

# 

# 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

# Example Input:

# 5

# 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

# PROGRAM:

# def distinct\_elements(arr):

# distinct\_set = set(arr)

# distinct\_list = list(distinct\_set)

# distinct\_list.sort()

# return distinct\_list

# 

# #Input

# n = int(input())

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

# #Output

# distinct = distinct\_elements(arr)

# print(" ".join(map(str, distinct)))

# 

**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.

Sample Test Cases

Test Case 1

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

# PROGRAM:

# x=[]

# for i in range(0,11):

# b=int(input())

# x.append(b)

# #a.sort()

# print("ITEM to be inserted:",x[-1],sep='')

# x.sort()

# print("After insertion array is:")

# for i in x:

# print(i)

# 

# 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 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.

# Sample Case 0

# 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 |
| --- | --- |
| 103 | 5 |
| 105 | 0 |
| 11 | 1 |

# PROGRAM:

# import math

# n = int(input())

# p = int(input())

# factors = []

# for i in range(1, int(math.sqrt(n)) + 1):

# if n % i == 0:

# factors.append(i)

# if i != n // i:

# factors.append(n // i)

# factors.sort()

# if p <= len(factors):

# print(factors[p - 1])

# 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

# Sample test case

# Sample input

# 2

# 2

# 1

# 3

# 5

# 7

# 2

# 4

# 6

# 8

# Sample Output

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

# PROGRAM:

# def zip\_lists(list1,list2):

# return [a+b for a,b in zip(list1,list2)]

# m=int(input())

# n=int(input())

# list1=[[int(input()) for \_ in range(n)]for \_ in range(m)]

# list2=[[int(input()) for \_ in range(n)]for \_ in range(m)]

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

# print(zipped\_list)

# 

# 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

# Sample Input 1

# 5

# 1

# 2

# 3

# 6

# 9

# 4

# 2

# 4

# 5

# 10

# Sample Output 1

# 1 2 3 4 5 6 9 10

# PROGRAM:

# n1=int(input())

# l1=[]

# for i in range(0,n1):

# a=int(input())

# l1.append(a)

# n2=int(input())

# l2=[]

# for i in range(0,n2):

# a=int(input())

# l2.append(a)

# l3=[]

# l3.extend(l1)

# l3.extend(l2)

# a=list(set(l3))

# a.sort()

# for i in a:

# print(i,end=' ')

# 

# 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.

# 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.

# PROGRAM:

# n = int(input())

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

# element\_to\_search = int(input())

# locations = []

# occurrences = 0

# for i in range(len(arr)):

# if arr[i] == element\_to\_search:

# locations.append(i + 1)

# occurrences += 1

# # Output

# if occurrences == 0:

# print(f"{element\_to\_search} is not present in the array.")

# else:

# 

# for loc in locations:

# print(f"{element\_to\_search} is present at location {loc}.")

# print(f"{element\_to\_search} is present {occurrences} times in the array.")

# 

# 

# Ex. No. : 5.10 Date:

# Register No.: Name:

Find the intersection of two sorted arrays.

OR in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays.

Input Format

The first line contains T, the number of test cases. Following T lines contain:

1.      Line 1 contains N1, followed by N1 integers of the first array

2.      Line 2 contains N2, followed by N2 integers of the second array

Output Format

The intersection of the arrays in a single line

Example

Input:

1

3 10 17 57

6 2 7 10 15 57 246

Output:

10 57

Input:

1

7

1

2

3

3

4

5

6

2

1

6

Output:

1 6

**For example:**

| **Input** | **Result** |
| --- | --- |
| 1  3  10  17  57  6  2  7  10  15  57  246 | 10 57 |
| 1  7  1  2  3  3  4  5  6  2  1  6 | 1 6 |

# PROGRAM:

# t=int(input())

# l1=list()

# while(t!=0):

# n1=int(input())

# l1=[]

# l2=[]

# for i in range(0,n1):

# a=int(input())

# l1.append(a)

# n2=int(input())

# for i in range(0,n2):

# a=int(input())

# l2.append(a)

# t=t-1

# c=set(l1)

# d=set(l2)

# e=list(c.intersection(d))

# e.sort()

# for i in e:

# print(i,end=' ')

# print('\n')

# 