

## WEEK -6 PYTHON PROGRAMMING

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

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.

Example

Input

1  
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

SOLUTION:

```
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')
```

2. 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  $i$ th 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  $[0, 10^5]$ .

SOLUTION:

```
n=int(input())
```

```
a=list(map(int, input().split()))
```

```
b=list(map(int, input().split()))
```

```
index_map={}
```

```
for i, num in enumerate(b):
```

```
    index_map[num]=i
```

```
mapping = [index_map[num] for num in a]
```

```
print(*mapping)
```

3. 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"

Sample Test Case

Input

7

1

2

3

0

4

5

6

Output

True

SOLUTION:

```
n = int(input())
```

```
lst = [int(input()) for _ in range(n)]
```

```
increasing = all(lst[i] < lst[i+1] for i in range(len(lst)-1))
```

```
decreasing = all(lst[i] > lst[i+1] for i in range(len(lst)-1))
```

if increasing or decreasing:

```

    print("True")
else:
    for i in range(len(lst)):
        temp = lst[:i] + lst[i+1:]
        if all(temp[j] < temp[j+1] for j in range(len(temp)-1)) or all(temp[j] > temp[j+1] for j in
range(len(temp)-1)):
            print("True")
            break
    else:
        print("False")

```

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

Example

$n = 20$

$p = 3$

The factors of 20 in ascending order are {1, 2, 4, 5, 10, 20}. Using 1-based indexing, if  $p = 3$ , then 4 is returned. If  $p > 6$ , 0 would be returned.

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.

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  $p = 3^{\text{rd}}$  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.

SOLUTION:

```
a=int(input())
p=int(input())
f=[]
f.append(1)
j=1
for i in range(1,a+1):
    if a%i==0:
        # print('1')
        f.append(i)
        j=j+1
#for j in range (1,j):
    #print(f[j])
#print(j,p)
if j<=p:
    print('0')
else:
    e=f[p]
    print(e)
```

5. 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

SOLUTION:

```
a=int(input())
```

```
l1=[]
```

```
d=0
```

```
for i in range (0,a):
```

```
    b=int(input())
```

```

l1.append(b)
c=int(input())
d=a+c
for i in range(a,d):
    b=int(input())
    l1.append(b)
l1.sort()
l1=set(l1)
l1=list(l1)
d=len(l1)
l1.sort()
for i in range(0,d):
    print(l1[i],end=' ')

```

6.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	10 57
3	
10	
17	
57	
6	
2	
7	
10	
15	
57	
246	

1	1 6
7	
1	
2	
3	
3	
4	
5	
6	
2	
1	
6	

SOLUTION:

```
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')
```

7. 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]]

SOLUTION:

```
m=int(input())
```

```
n=int(input())
```

```
l1=[]
```

```
l2=[]
```

```
c=1
```

```
for i in range(0,m*n*2,2):
```

```
    a=int(input())
```

```
    b=int(input())
```

```
    if c%2!=0:
```

```
        l1.append(a)
```

```
        l1.append(b)
```

```
    else:
```

```
        l2.append(a)
```

```
        l2.append(b)
```

```
    c=c+1
```

```
l3=[]
```

```
l3.append(l1)
```

```
l3.append(l2)
```

```
print(l3)
```

8.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

SOLUTION:

```
a=[]
```

```
for i in range(0,11):
```

```
    b=int(input())
```

```
    a.append(b)
```

```
print("ITEM to be inserted:",a[-1],sep="")
```

```
a.sort()
```

```
print("After insertion array is:")
```

```
for i in a:
```

```
    print(i)
```

9. 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

- $3 \leq n \leq 10^5$
- $1 \leq \text{arr}[i] \leq 2 \times 10^4$ , 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  $\text{arr}$ .

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

#### 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,  $\text{arr}[2]=3$  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

SOLUTION:

```
a=int(input())
```

```
s=[]
```

```
for i in range(a):
```

```
    b=int(input())
```

```
    s.append(b)
```

```
l=0
```

```
r=0
```

```
for i in range(0,a):
```

```
    for j in range(0,i):
```

```
        l=l+s[j]
```

```
    for k in range(i+1,a):
```



```
    r=r+s[k]
if(l==r):
    print(i)
l=r=0
```