

[Dashboard](#) / [My courses](#) / [CS23221-PPL-2023](#) / [Experiments based on Lists and its operations.](#) / [Week6 Coding](#)

Started on	Friday, 3 May 2024, 7:29 PM
State	Finished
Completed on	Monday, 13 May 2024, 6:54 PM
Time taken	9 days 23 hours

Question 1

Correct

Marked out of 1.00

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

Answer: (penalty regime: 0 %)

```

1  n = int(input(""))
2  list1 = [int(input()) for _ in range(n)]
3
4  def is_strictly_increasing(lst):
5      count = 0
6      for i in range(1, len(lst)):
7          if lst[i] <= lst[i - 1]:
8              count += 1
9          if count > 1:
10             return False
11         # Check if removing the current or previous element helps
12         if i == 1 or lst[i] > lst[i - 2]:
13             continue
14         elif i < len(lst) - 1 and lst[i + 1] > lst[i - 1]:
15             continue
16         else:
17             return False
18     return True
19
20 def is_strictly_decreasing(lst):
21     reversed_lst = lst[::-1]
22     return is_strictly_increasing(reversed_lst)
23
24 if is_strictly_increasing(list1) or is_strictly_decreasing(list1):
25     print("True")
26 else:
27     print("False")

```

	Input	Expected	Got	
✓	7 1 2 3 0 4 5 6	True	True	✓
✓	4 2 1 0 -1	True	True	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 1.00

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 `arr`.

Each of the next n lines contains an integer, `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, `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

Answer: (penalty regime: 0 %)

```
1 n = int(input())
2 arr = []
3 for i in range(n):
4     arr.append(int(input()))
5 for i in range(n):
6     left_sum = sum(arr[:i])
7     right_sum = sum(arr[i+1:])
8     if left_sum == right_sum:
9         print(i)
10        break
11
```

	Input	Expected	Got	
✓	4 1 2 3 3	2	2	✓
✓	3 1 2 1	1	1	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 1.00

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

Answer: (penalty regime: 0 %)

```
1 a=list()
2 for i in range(10):
3     b=int(input())
4     a.append(b)
5 c=int(input())
6 print("ITEM to be inserted:",c,sep="")
7 print("After insertion array is:")
8 a.append(c)
9 a=list(set(a))
10 a.sort()
11 for i in a:
12     print(i)
```

	Input	Expected	Got	
✓	1 3 4 5 6 7 8 9 10 11 2	ITEM to be inserted:2 After insertion array is: 1 2 3 4 5 6 7 8 9 10 11	ITEM to be inserted:2 After insertion array is: 1 2 3 4 5 6 7 8 9 10 11	✓

	Input	Expected	Got	
✓	11 22 33 55 66 77 88 99 110 120 44	ITEM to be inserted:44 After insertion array is: 11 22 33 44 55 66 77 88 99 110 120	ITEM to be inserted:44 After insertion array is: 11 22 33 44 55 66 77 88 99 110 120	✓

Passed all tests! ✓

Question 4

Correct

Marked out of 1.00

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.

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.

Sample Case 2**Sample Input 2**

1

1

Sample Output 2

1

Explanation 2

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

For example:

Input	Result
10 3	5
10 5	0
1 1	1

Answer: (penalty regime: 0 %)

```

1 a=int(input())
2 b=int(input())
3 c=list()
4 for i in range(1,a+1):
5     if(a%i==0):
6         c.append(i)
7 if(b<=len(c)):
8     print(c[b-1])
9 else:
10    print(0)

```

	Input	Expected	Got	
✓	10 3	5	5	✓
✓	10 5	0	0	✓
✓	1 1	1	1	✓

Passed all tests! ✓

Question **5**

Correct

Marked out of 1.00

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

Answer: (penalty regime: 0 %)

```

1 T=int(input())
2 n1=int(input())
3 list1=[]
4 list2=[]
5
6 for i in range(0,n1):
7     list1.append(int(input()))
8 n2=int(input())
9 for i in range(0,n2):
10    list2.append(int(input()))
11 for i in range(0,n1):
12     for j in range(0,n2):
13         if list1[i]==list2[j]:
14             print(list1[i],end=" ")

```

	Input	Expected	Got	
✓	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✓
✓	1 7 1 2 3 3 4 5 6 2 1 6	1 6	1 6	✓

Passed all tests! ✓

Question **6**

Correct

Marked out of 1.00

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

Answer: (penalty regime: 0 %)

```
1 t=int(input())
2 for i in range(t):
3     n=int(input())
4     c=[]
5     for i in range(n):
6         d=int(input())
7         c.append(d)
8     k=int(input())
9     flag=0
10    for i in range(len(c)):
11        for j in range(i+1,len(c)):
12            if abs(c[i]-c[j])==k and i!=j:
13                flag=1
14                break
15        if flag==1:
16            break
17    print(1 if flag else 0)
18
```

	Input	Expected	Got	
✓	1 3 1 3 5 4	1	1	✓
✓	1 3 1 3 5 99	0	0	✓

Passed all tests! ✓

Question 7

Correct

Marked out of 1.00

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 $[0, 10^5]$.

Answer: (penalty regime: 0 %)

```
1 a=int(input())
2 c=list(dict.fromkeys(list(map(int,input().split()))))
3 d=list(dict.fromkeys(list(map(int,input().split()))))
4 for i in c:
5     print(d.index(i),end=" ")
```


	Input	Expected	Got	
✓	5 12 28 46 32 50 50 12 32 46 28	1 4 3 2 0	1 4 3 2 0	✓

Passed all tests! ✓

Question **8**

Correct

Marked out of 1.00

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.

Answer: (penalty regime: 0 %)

```
1 a=int(input())
2 b=list()
3 for i in range(a):
```

```
4 | c=int(input())
5 | b.append(c)
6 | d=int(input())
7 | e=len(b)
8 | x=0
9 | for i in range(e):
10 |     if(d==b[i]):
11 |         print(d,"is present at location {}".format(i+1))
12 |         x+=1
13 | if(x==0):
14 |     print(d,"is not present in the array.")
15 | else:
16 |     print(d,"is present",x,"times in the array.")
17 |
18 |
19 |
```

	Input	Expected	Got	
✓	4 5 6 5 7 5	5 is present at location 1. 5 is present at location 3. 5 is present 2 times in the array.	5 is present at location 1. 5 is present at location 3. 5 is present 2 times in the array.	✓
✓	5 67 80 45 97 100 50	50 is not present in the array.	50 is not present in the array.	✓

Passed all tests! ✓

Question **9**

Correct

Marked out of 1.00

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

Answer: (penalty regime: 0 %)

```
1 a=int(input())
2
3 x=list()
4 for i in range(a):
5     b=int(input())
6     x.append(b)
7 c=int(input())
8 for i in range(c):
9     d=int(input())
10    x.append(d)
11 x=list(set(x))
12 x.sort()
13 for i in x:
14     print(i,end=" ")
```

	Input	Expected	Got	
✓	5 1 2 3 6 9 4 2 4 5 10	1 2 3 4 5 6 9 10	1 2 3 4 5 6 9 10	✓
✓	7 4 7 8 10 12 30 35 9 1 3 4 5 7 8 11 13 22	1 3 4 5 7 8 10 11 12 13 22 30 35	1 3 4 5 7 8 10 11 12 13 22 30 35	✓

Passed all tests! ✓

Question 10

Correct

Marked out of 1.00

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]]

Answer: (penalty regime: 0 %)

```
1 # Input for row size and column size
2 m = int(input())
3 n = int(input())
4
5 # Input for the first list of lists
6 list1 = []
7 for i in range(m):
8     row = []
9     for j in range(n):
10         row.append(int(input()))
11     list1.append(row)
12
13 # Input for the second list of lists
14 list2 = []
15 for i in range(m):
16     row = []
17     for j in range(n):
18         row.append(int(input()))
19     list2.append(row)
20
21 # Zipping the two lists of lists
22 zipped_list = []
23 for i in range(m):
24     zipped_list.append(list1[i] + list2[i])
25
26 print(zipped_list)
27
```

	Input	Expected	Got	
✓	2 2 1 2 3 4 5 6 7 8	[[1, 2, 5, 6], [3, 4, 7, 8]]	[[1, 2, 5, 6], [3, 4, 7, 8]]	✓

Passed all tests! ✓

[◀ Week5_Coding](#)

Jump to...

[Week7_MCQ ▶](#)