resource url?
QUESTION ----------------
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
------------------ANSWER
```
a= int(input())
bef= int(input())
c= 0
if a== 4:
    print(True)
else :
    for i in range(a- 1):
        now= int(input())
        if bef>=now:
            c += 1
            if c> 1:
                print(False)
                break
    else:
        print(True)
```

QUESTION ----------------
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
.
$3 \leq n \leq 105$
.
$1 \leq arr[i] \leq 2 \times 104$, 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.

```
------------------ANSWER
a= int(input())
val= []
for i in range(a):
    val.append(int(input()))
piv= 1
while piv <a-1:
    l= sum(val[:piv])
    r= sum(val[piv+1:])
    if l== r:
        print(piv)
        break
    piv+=1
```

```
------------------------------
QUESTION ----------------
Find the intersection of two sorted arrays.
------------------ANSWER
val= int(input())

for i in range(val):
    l1,l2= [], []

    n1= int(input())
    for i in range(n1):
        l1.append(int(input()))
    n2= int(input())
    for i in range(n2):
        l2.append(int(input()))

    intsec= []

    a=b=0
    while a< n1 and b< n2:
        p,q= l1[a], l2[b]
        if p==q:
            intsec.append(p)
            a+= 1
            b+= 1
        elif p<q:
            a+= 1
        elif p>q:
            b+= 1


    print(*intsec)
------------------------------
QUESTION ----------------
```

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.

------------------ANSWER

```
def yieldFactors(c):
    for i in range(1,c+1):
        if c%i == 0:
            yield i

fact= list(yieldFactors(int(input())))
a= int(input())
if a<= len(fact):
    print(fact[a-1])
else:
    print(0)
```

QUESTION ----------------

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.

------------------ANSWER

```
from bisect import bisect

nums = [int(input()) for i in range(10)]
num = int(input())

nums.insert(bisect(nums, num), num)
print(f"ITEM to be inserted:{num}")
print("After insertion array is:")
print(*nums, sep='\n')
```

QUESTION ----------------

Write a Python program to Zip two given lists of lists.

------------------ANSWER

```
list1 = []
list2 = []
m, n = int(input()), int(input())
for i in range(m):
    child = []
    for i in range(n):
        child.append(int(input()))
    list1.append(child)

for i in range(m):
    child = []
    for i in range(n):
        child.append(int(input()))
    list2.append(child)


first, last  = [*list1[0], *list2[0]], [*list1[1], *list2[1]]
print([first, last])
```

QUESTION ----------------

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.

------------------ANSWER

```
v=int(input())
Arr1=input()
A=[]
```

```
n=''
for ch in Arr1:
    if ch==' ':
        A.append(int(n))
        n=''
    else:
        n+=ch
if n!='':
    A.append(int(n))
Arr2=input()
B=[]
n=''
for ch in Arr2:
    if ch==' ':
        B.append(int(n))
        n=''
    else:
        n+=ch
if n!='':
    B.append(int(n))
D={}
for i in range(len(B)):
    D[B[i]]=i
L=[]
for n in A:
    L.append(D[n])
for i in L:
    print(i,end=" ")
```

QUESTION ----------

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.

----------ANSWER

```
a= int(input())

def solve(nums, k):
    for i in nums:
        complement = abs(i - k)
        if complement in nums:
            return 1
    return 0

while a:
    a-= 1
    n = int(input())
    nums = [int(input()) for i in range(n)]
    k = int(input())
    print(solve(nums, k))
```

QUESTION ----------

Output is a
merged array without duplicates.

----------ANSWER

```
n1=int(input())
a1=[]
for i in range(n1):
    a1.append(int(input()))
n2=int(input())
a2=[]
for i in range(n2):
    a2.append(int(input()))
mergedarr=a1.copy()
for j in a2:
```

```
    if j not in mergedarr:
        mergedarr.append(j)
mergedarr.sort()
for i in mergedarr:
    print(i,end=" ")
```

QUESTION ----------

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.

----------ANSWER

```
n=int(input())
a=[]
for i in range(n):
    ele=int(input())
    a.append(ele)
k=int(input())
L=[]
c=0
for i in range (len(a)):
    if a[i]==k:
        L.append(i+1)
        c+=1
if c==0:
    print(f"{k} is not present in the array.")
else:
    for j in L:
        print(f"{k} is present at location {j}.")
    print(f"{k} is present {c} times in the array.")
```
