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
· $3 \le n \le 10^5$
· $1 \le arr[i] \le 2 \times 10^4$, where $0 \le i < n$
·   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 $0 \le i < n$.

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

Code:

```python
n= int(input())
arr = [int(input()) for _ in range(n)]
total_sum = sum(arr)
left_sum = 0
for i in range(len(arr)):
    total_sum -= arr[i]
    if left_sum == total_sum:
        print(i)
        break
    left_sum += arr[i]
```

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	Result
1 3 1 3 5 4	1
1 3 1 3 5 99	0

Code:

```
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')
```

Count Elements

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

Sample Test Cases
Test Case 1
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

Code:

```python
n=int(input())
freq={}
for _ in range(n):
    num = int(input())
    if num in freq:
        freq[num] += 1
    else:
        freq[num] = 1
for key, value in freq.items():
    print(f"{key} occurs {value} times")
```

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

Code:

```
def printDistinct(arr):
    distinct = []
    for num in arr:
        if num not in distinct:
            distinct.append(num)
    print(*distinct)
n = int(input())
arr = []
for _ in range(n):
  arr.append(int(input()))
printDistinct(arr)
```

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

Code:

```
a=[]
for x in range(0,11):
    b=int(input())
    a.append(b)
m=len(a)
print("ITEM to be inserted:",a[m-1],sep='')
a.sort()
print("After insertion array is:"
for x in a:
    print(x)
```

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

$1 \leq n \leq 10^{15}$

$1 \leq p \leq 10^{9}$

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 = 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

Code:

```
a=int(input())
for i in range(1,a+1):
    if(a%i==0):
        print(i,end=" ")
```

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]]

Code:

```python
m=int(input())
n=int(input())
l1 = [[int(input()) for _ in range(n)] for _ in range(m)]
l2 = [[int(input()) for _ in range(n)] for _ in range(m)]
zipp = [a + b for a, b in zip(l1, l2)]
print(zipp)
```

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

Code:

```python
n=int(input())
l=[]
for x in range(n):
    a=int(input())
    l.append(a)
o=int(input())
m=[]
for x in range(o):
    a=int(input())
    m.append(a)
l.extend(m)
l=set(l)
l=list(l)
l.sort()
for y in l:
    print(y,end='')
```

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.

Code:
```
a=[]
c=0
while(True):
  try:
    n=str(input())
    a.append(int(n)
except EOFError:
    break
if a[-1] in a[:-1:]:
  for i in range(len(a)-1):
    if a[-1]==a[i]:
        print(a[-1],"is present at location %d."%i)
        c=c+
  print(a[-1],"is present %d times in the array."%c)
else:
    print(a[-1],"is not present in the array.")
```

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"

Sample Test Case

Input

7

1

2

3

0

4

5

6

Output

True

Code:
```python
n = int(input())
lst = [int(input()) for _ in range(n)]
strict_inc = all(lst[i] < lst[i+1] for i in range(len(lst)-1))
strict_dec = all(lst[i] > lst[i+1] for i in range(len(lst)-1))
if strict_inc or strict_dec:
    print("True")
else:
    for i in range(len(lst)):
        temp_lst = lst[:i] + lst[i+1:]
        temp_inc = all(temp_lst[j] < temp_lst[j+1] for j in range(len(temp_lst)-1))
        temp_dec = all(temp_lst[j] > temp_lst[j+1] for j in range(len(temp_lst)-1))
        if temp_inc or temp_dec:
            print("True")
            break
    else:
        print("False")
```