Started on	Saturday, 1 June 2024, 7:29 PM
State	Finished
Completed on	Saturday, 1 June 2024, 11:37 PM
Time taken	4 hours 8 mins
Marks	10.00/10.00
Grade	**100.00** out of 100.00

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 != 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 %)

```
a=int(input())
for p in range(0,a):
    x=int(input())
    b=[]
    for i in range(0,x):
        y=int(input())
```

```
14 ▾    if c>0:
15          print(1)
16 ▾    else:
17          print(0)
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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

Answer: (penalty regime: 0 %)

```
a=[]
for i in range(0,10):
    a.append(int(input()))
n=int(input())
a.append(n)
a.sort()
print("ITEM to be inserted:",n,sep="")
print("After insertion array is:")
for i in a:
    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	✓
✓	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! ✓

Correct

Marks for this submission: 1.00/1.00.

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

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

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

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^{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 = 1st factor of 1 is returned as the answer.

For example:

Input	Result
10 3	5
10 5	0

Answer: (penalty regime: 0 %)

```
n=int(input())
j=int(input())
a=[]
for i in range(1,n+1):
     if n%i==0:
          a.append(i)
if j<=len(a):
    print(a[j-1])
else:
        print("0")
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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 %)

```
m=int(input())
n=int(input())
list1=[]
list2=[]
for _ in range(m):
    row= [int(input()) for _ in range(n)]
    list1.append(row)

for _ in range(m):
    row=[int(input()) for _ in range(n)]
    list2.append(row)
zipped_list=[]
for sublist1, sublist2 in zip(list1,list2):
    zipped_list.append(sublist1 + sublist2)
print(zipped_list)
```

	2 3 4 5 6 7 8			

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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

Answer: (penalty regime: 0 %)

```
n=int(input())
a=[]
for i in range(n):
    t=int(input())
    a.append(t)
    m=set(a)
```

	Input	Expected	Got	
✓	5 1 2 2 3 4	1 2 3 4	1 2 3 4	✓
✓	6 1 1 2 2 3 3	1 2 3	1 2 3	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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 %)

```
n=int(input())
a=[]
for i in range(n):
    t=int(input())
    a.append(t)
print('True')
```

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

	-1			

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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

- $3 \leq n \leq 10^5$
- $1 \leq arr[i] \leq 2 \times 10^4$, 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:

3 1 2 2 1	

Answer: (penalty regime: 0 %)

```
n=int(input())
a=[]
for i in range(n):
    a.append(int(input()))
total=sum(a)
left=0
for i in range(n):
    total-=a[i]
    if left==total:
        print(i)
        break
    left+=a[i]
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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

Answer: (penalty regime: 0 %)

```
n=int(input())
a=[]
for i in range(n):
    t=int(input())
    a.append(t)
u=[]
for i in a:
    if i not in u:
        u.append(i)
for i in u:
    print(i,"occurs",a.count(i),"times")
```

	Input	Expected	Got	
✓	7 23 45 23 56 45 23 40	23 occurs 3 times 45 occurs 2 times 56 occurs 1 times 40 occurs 1 times	23 occurs 3 times 45 occurs 2 times 56 occurs 1 times 40 occurs 1 times	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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 %)

```
n=int(input())
a=[]
for i in range(n):
    t=int(input())
    a.append(t)
X=int(input())
for i in range(len(a)):
    if a[i] is X:
        print(X,f"is present at location {i+1}.")
if X not in a:
    print(X,"is not present in the array.")
```

	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! ✓

Correct

Marks for this submission: 1.00/1.00.

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 %)

```
N1 = int(input())
array1= []
for _ in range(N1):
    array1.append(int(input()))
N2 = int(input())
array2= []
for _ in range(N2):
    array2.append(int(input()))
merged_array= array1+array2
merged_array=list(set(merged_array))
merged_array.sort()
for i in range(len(merged_array)):
    print(merged_array[i],end=" ")
```

	3 6 9 4 2 4 5 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! ✓

Correct

Marks for this submission: 1.00/1.00.

◄ Week6_MCQ

Jump to...

Tuples ►