Time taken	7 hours 32 mins
Marks	10.00/10.00
Grade	**100.00** out of 100.00

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

	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.

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 %)

```
a = int(input())
b =[]
for i in range(a):
    e = int(input())
    b.append(e)
c = int(input())
d =[]
for i in range(c):
    f = int(input())
    d.append(f)
b.extend(d)
b.sort()
g =[]
for i in b:
    if i not in g:
        g.append(i)
for i in g:
    print(i,end=" ")
```

	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.

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

110
120

Answer: (penalty regime: 0 %)

```
li1=[int(input()) for i in range(10)]
b=int(input())
li1.append(b)
li1.sort()
print(f'ITEM to be inserted:{b}')
print("After insertion array is:  ")
for i in li1:
    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	✓

	120 44	88 99 110 120	88 99 110 120	

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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())
l1=[]
l2=[]
l3=[]
for i in range(m*n*n):
    l1.append(int(input()))
for i in range(len(l1)):
    if i<2 or 4<=i<=5:
        l2.append(l1[i])
    else:
        l3.append(l1[i])
f=[]
f.append(l2)
f.append(l3)
print(f)
```

7				
8				

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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.

1	
1	

Answer: (penalty regime: 0 %)

```
n=int(input())
x=[]
for _ in range(n):
    x.append(int(input()))
t=sum(x)
l=0
r=t -x[0]
for i in range(1,n):
    l+=x[i-1]
    r-=x[i]
    if l==r:
        print(i)
        break
```

	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.

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

```
    while end < n:
        if start==end:
            end +=1
        elif a[end]-a[start]==k:
            re.append(1)
            found=True
            break
        elif a[end]-a[start]<k:
            end +=1
        else:
            start+=1
    if not found:
        re.append(0)
for result in re:
    print(result)
```

	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.

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())
li1=[]
for i in range(n):
```

```
    print(f'{search_num} 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.

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 %)

```
e = int(input())
a =[]
for i in range(e):
    b = int(input())
    a.append(b)
c =[]
for i in a:
    if i not in c:
        c.append(i)
for i in c:
    d = a.count(i)
    print(f'{i} occurs {d} times')

```

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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())
li=[]
for i in range(n):
    li.append(int(input()))
li.sort()
res=all(i< j for i,j in zip(li,li[1:]))
print(res)
```

✓	4 1 1 0 -1	True	True	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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 = 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:

Answer: (penalty regime: 0 %)

```
a = int(input())
b = int(input())
c = []
for i in range(1,a+1):
    if a%i==0:
        c.append(i)
if len(c)>=b:
    print(c[b-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.

◄ Week6_MCQ

Jump to...

Tuples ►