1. Given an listof integers, sort the array in ascending order using the *Bubble Sort* algorithm above. Once sorted, print the following three lines:

1.      List is sorted in numSwaps swaps., where numSwaps is the number of swaps that took place.

2.      First Element: firstElement, the  *first* element in the sorted list.

3.      Last Element: lastElement, the *last* element in the sorted list.

For example, given a worst-case but small array to sort: a=[6,4,1]. It took  3 swaps to sort the array. Output would be

Array is sorted in 3 swaps.

First Element: 1

Last Element: 6

**Input Format**

The first line contains an integer,n , the size of the list a .  
The second line contains  n,  space-separated integers a[i].

**Constraints**

·         2<=n<=600

·         1<=a[i]<=2x106.

**Output Format**

You must print the following three lines of output:

1.      List is sorted in numSwaps swaps., where numSwaps is the number of swaps that took place.

2.      First Element: firstElement, the  *first* element in the sorted list.

3.      Last Element: lastElement, the *last* element in the sorted list.

**Sample Input 0**

3

1 2 3

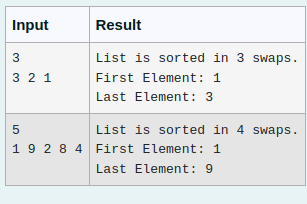
**Sample Output 0**

List is sorted in 0 swaps.

First Element: 1

Last Element: 3

For Example



Answer

def bubble\_sort(arr):

n = len(arr)

num\_swaps = 0

for i in range(n):

for j in range(0, n - i - 1):

if arr[j] > arr[j + 1]:

arr[j], arr[j + 1] = arr[j + 1], arr[j]

num\_swaps += 1

return num\_swaps

n = int(input())

arr = list(map(int, input().split()))

num\_swaps = bubble\_sort(arr)

print("List is sorted in", num\_swaps, "swaps.")

print("First Element:", arr[0])

print("Last Element:", arr[-1])

2. Given an list, find peak element in it. A peak element is an element that is greater than its neighbors.

An element a[i] is a peak element if

A[i-1] <= A[i] >=a[i+1] for middle elements. [0<i<n-1]

A[i-1] <= A[i] for last element [i=n-1]

A[i]>=A[i+1] for first element [i=0]

**Input Format**

The first line contains a single integer n , the length of A .  
The second line contains n space-separated integers,A[i].

**Output Format**

**Print** peak numbers separated by space.

**Sample Input**

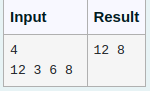
5

8 9 10 2 6

**Sample Output**

10 6

For Example



Answer

def find\_peak\_elements(arr):

n = len(arr)

peak\_elements = []

if n > 1 and arr[0] >= arr[1]:

peak\_elements.append(arr[0])

for i in range(1, n - 1):

if arr[i - 1] <= arr[i] >= arr[i + 1]:

peak\_elements.append(arr[i])

if n > 1 and arr[-1] >= arr[-2]:

peak\_elements.append(arr[-1])

return peak\_elements

n = int(input())

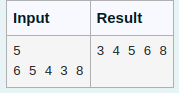
arr = list(map(int, input().split()))

peak\_elements = find\_peak\_elements(arr)

print(\*peak\_elements)

3. Write a Python program to sort a list of elements using the merge sort algorithm.

For example:



Answer

def merge\_sort(arr):

if len(arr) > 1:

mid = len(arr) // 2

left\_half = arr[:mid]

right\_half = arr[mid:]

merge\_sort(left\_half)

merge\_sort(right\_half)

i = j = k = 0

while i < len(left\_half) and j < len(right\_half):

if left\_half[i] < right\_half[j]:

arr[k] = left\_half[i]

i += 1

else:

arr[k] = right\_half[j]

j += 1

k += 1

while i < len(left\_half):

arr[k] = left\_half[i]

i += 1

k += 1

while j < len(right\_half):

arr[k] = right\_half[j]

j += 1

k += 1

def print\_list(arr):

for element in arr:

print(element, end=" ")

print()

x=int(input())

y=[int(i) for i in input().split()]

merge\_sort(y)

for i in y:

print(i,end=" ")

4. o find the frequency of numbers in a list and display in sorted order.

**Constraints:**

1<=n, arr[i]<=100

**Input:**

1 68 79 4 90 68 1 4 5

**output:**

 1 2

 4 2

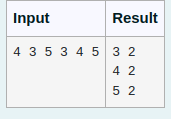
 5 1

 68 2

 79 1

90 1

For example:



Answer

x=[int(i) for i in input().split()]

y=sorted(list(set(x)))

for i in y:

print(i,x.count(i))

5. An list contains N numbers and you want to determine whether two of the numbers sum to a given number K. For example, if the input is 8, 4, 1, 6 and K is 10, the answer is yes (4 and 6). A number may be used twice.

**Input Format**

The first line contains a single integer n , the length of list

The second line contains n space-separated integers, list[i].

The third line contains integer k.

**Output Format**

Print Yes or No.

**Sample Input**

7

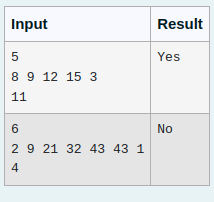
0 1 2 4 6 5 3

1

**Sample Output**

Yes

For example:



Answer

n = int(input())

arr = [int(x) for x in input().split()]

k = int(input())

complements = set()

found = False

for num in arr:

if (k - num) in complements:

found = True

break

complements.add(num)

if found:

print("Yes")

else:

print("No")