**10 - Searching & Sorting**

**Ex. No. : 10.1 Date:25/5/24**

**Register No.: 231501015 Name: Aniruth S V**

**Merge Sort**

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

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  6 5 4 3 8 | 3 4 5 6 8 |

**Program:**

a = int(input())

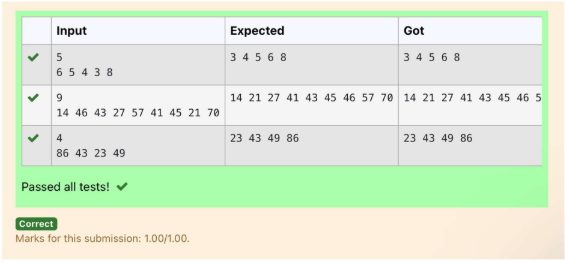
b = list(input().split(" "))

b.sort()

for i in b:

print(i,end=" ")

**Output:**

****

**Ex. No. : 10.2 Date:25/5/24**

**Register No.: 231501015 Name: Aniruth S V**

**Bubble Sort**

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](http://118.185.187.137/moodle/mod/resource/view.php?id=1068) 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](http://118.185.187.137/moodle/mod/resource/view.php?id=1068).

3.      Last Element: lastElement, the *last* element in the sorted [list](http://118.185.187.137/moodle/mod/resource/view.php?id=1068).

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](http://118.185.187.137/moodle/mod/resource/view.php?id=1068) 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](http://118.185.187.137/moodle/mod/resource/view.php?id=1068) 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](http://118.185.187.137/moodle/mod/resource/view.php?id=1068).

3. Last Element: lastElement, the *last* element in the sorted [list](http://118.185.187.137/moodle/mod/resource/view.php?id=1068).

**Sample Input 0**

3

1 2 3

**Sample Output 0**

[List](http://118.185.187.137/moodle/mod/resource/view.php?id=1068) is sorted in 0 swaps.

First Element: 1

Last Element: 3

**For example:**

| **Input** | **Result** |
| --- | --- |
| 3  3 2 1 | List is sorted in 3 swaps.  First Element: 1  Last Element: 3 |
| 5  1 9 2 8 4 | List is sorted in 4 swaps.  First Element: 1  Last Element: 9 |

**Program:**

n=input()

arr=input().split()

arr1=[]

for i in arr:

arr1.append(int(i))

l = len(arr1)

nswaps = 0

for i in range(l):

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

if arr1[j] > arr1[j+1]:

arr1[j], arr1[j+1] = arr1[j+1], arr1[j]

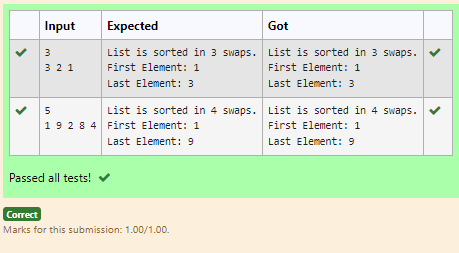
nswaps +=1

print("List is sorted in {} swaps.".format(nswaps))

print("First Element: {}".format(arr1[0]))

print("Last Element: {}".format(arr1[-1]))

**Output:**

****

**Ex. No. : 10.3 Date:25/5/24**

**Register No.: 231501015 Name: Aniruth S V**

**Peak Element**

Given an [list](http://118.185.187.137/moodle/mod/resource/view.php?id=1068), 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:**

| **Input** | **Result** |
| --- | --- |
| 4  12 3 6 8 | 12 8 |

**Program:**

def find\_peaks(n, A):

peaks = []

if n == 1 or A[0] >= A[1]:

peaks.append(A[0])

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

if A[i] >= A[i - 1] and A[i] >= A[i + 1]:

peaks.append(A[i])

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

peaks.append(A[-1])

return peaks

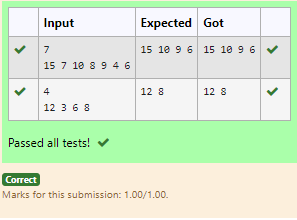
n = int(input().strip())

A = list(map(int, input().strip().split()))

peaks = find\_peaks(n, A)

print(" ".join(map(str, peaks)))

**Output:**

****

**Ex. No. : 10.4 Date:25/5/24**

**Register No.: 231501015 Name: Aniruth S V**

**Binary Search**

Write a Python program for binary search.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 1 2 3 5 8  6 | False |
| 3 5 9 45 42  42 | True |

**Program:**

def binary\_search(arr, target):

left, right = 0, len(arr) - 1

while left <= right:

mid = (left + right) // 2

if arr[mid] == target:

return True

elif arr[mid] < target:

left = mid + 1

else:

right = mid - 1

return False

if \_\_name\_\_ == "\_\_main\_\_":

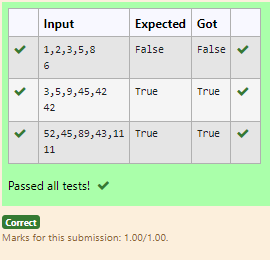
arr1 = list(map(int, input().strip().split(',')))

target1 = int(input().strip())

result1 = binary\_search(sorted(arr1), target1)

print(result1)

**Output:**

****

**Ex. No. : 10.5 Date:25/5/24**

**Register No.: 231501015 Name: Aniruth S V**

**Frequency of Elements**

To 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:**

| **Input** | **Result** |
| --- | --- |
| 4 3 5 3 4 5 | 3 2  4 2  5 2 |

**Program:**

num=input()

l=num.split()

nl=[]

for i in l:

nl.append(int(i))

ul=[]

nl.sort()

for i in nl:

if i not in ul:

ul.append(i)

for i in ul:

print("{} {}".format(i,nl.count(i)))

**Output:**

