10-Searching&Sorting

Ex.No. : 10.1 **Date: 02-05-2024**

RegisterNo.: 2116231501072 **Name: Kanishka P**

MergeSort

WriteaPythonprogramtosortalistofelementsusingthemergesortalgorithm.

Forexample:

Input	Result
5 65438	34568

Program:

```
a=int(input())
l=[]
l.extend(input().split())
for i in range(a-1):
    for j in range(a-1):
        if(int(l[j])>int(l[j+1])):
            t=int(l[j])
            l[j]=int(l[j+1])
            l[j+1]=t
for i in range(a):
    print(int(l[i]),end="")
```

Output:

	Input	Expected	Got
✔	5 6 5 4 3 8	3 4 5 6 8	3 4 5 6 8
✔	9 14 46 43 27 57 41 45 21 70	14 21 27 41 43 45 46 57 70	14 21 27 41 43 45 46 5
✔	4 86 43 23 49	23 43 49 86	23 43 49 86

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Ex.No. : 10.2 **Date: 02-05-2024**

RegisterNo.: 2116231501072 **Name: Kanishka P**

BubbleSort

Givenanlistofintegers,sortthearrayinascendingorderusingthe*BubbleSort* algorithm above. Once sorted, print the following three lines:

1. Listis sorted in numSwapsswaps., wherenumSwapsisthe numberof swapsthat took place.
2. FirstElement:firstElement, the*first*elementinthesortedlist.
3. LastElement: lastElement,the*last*elementinthesortedlist.

Forexample,givenaworst-casebutsmallarraytosort:a=[6,4,1].Ittook3swapstosort the array. Output would be

Arrayissortedin3swaps.
First Element: 1
LastElement:6

InputFormat

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

Constraints

- 2<=n<=600
- 1<=a[i]<=2x10^6.

OutputFormat

Youmustprintthefollowingthreelinesofoutput:

1. ListissortedinnumSwapsswaps.,wherenumSwapsisthenumberofswapsthat took place.
2. FirstElement: firstElement, the*first*elementinthesortedlist.
3. LastElement: lastElement,the*last*elementinthesortedlist.

SampleInput0

3

123

SampleOutput0

Listissortedin0swaps.

First Element: 1

LastElement:3

Forexample:

Input	Result
3 321	Listissortedin3swaps. First Element: 1 LastElement:3
5 19284	Listissortedin4swaps. First Element: 1 LastElement:9

Program:

```
defbubble_sort(arr):
    n = len(arr)swaps
    = 0

    foriinrange(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j + 1]:
                # Swap elements
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
                swaps += 1

    returnswaps

# Input the size of the list
n = int(input())

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

# Perform bubble sort and count the number of swaps
num_swaps = bubble_sort(arr)
```

```
#Printthenumberofswaps
print("Listissortedin",num_swaps,"swaps.")

# Print the first element
print("FirstElement:",arr[0])

# Print the last element
print("LastElement:",arr[-1])
```

Output:

	Input	Expected	Got	
✔	3 3 2 1	List is sorted in 3 swaps. First Element: 1 Last Element: 3	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	List is sorted in 4 swaps. First Element: 1 Last Element: 9	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Ex.No. : 10.3 **Date: 02-05-2024**

RegisterNo.: 2116231501072 **Name: Kanishka P**

PeakElement

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

Anelementa[i]isapeakelementif

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]forfirstelement[i=0]

InputFormat

The first line containsa single integern ,the lengthof A .
Thesecondlinecontains nspace-separatedintegers,A[i].

OutputFormat

Printpeaknumbersseparatedbyspace.

SampleInput

5

891026

SampleOutput

106

Forexample:

Input	Result
4 12368	128

Program:

```
def find_peak(arr):
    peak_elements = []
```

```
    # Check for the first element
    if arr[0] >= arr[1]:
        peak_elements.append(arr[0])

    # Check for middle elements
    foriinrange(1,len(arr)-1):
        if arr[i - 1] <= arr[i] >= arr[i + 1]:
            peak_elements.append(arr[i])

    # Check for the last element
    if arr[-1] >= arr[-2]:
        peak_elements.append(arr[-1])

    return peak_elements

# Input the length of the list
n = int(input())

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

# Find peak elements and print the result
peak_elements = find_peak(arr)
print(*peak_elements)
```

Output:

	Input	Expected	Got	
✔	7 15 7 10 8 9 4 6	15 10 9 6	15 10 9 6	✔
✔	4 12 3 6 8	12 8	12 8	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Ex.No. : 10.4 **Date: 02-05-2024**

RegisterNo.: 2116231501072 **Name: Kanishka P**

BinarySearch

WriteaPythonprogramforbinarysearch.

Forexample:

Input	Result
12358 6	False
3594542 42	True

Program:

```
a=input().split(",")
b = input()
print(bina)
```

Output:

	Input	Expected	Got	
✔	1,2,3,5,8 6	False	False	✔
✔	3,5,9,45,42 42	True	True	✔
✔	52,45,89,43,11 11	True	True	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Ex.No. : 10.5 **Date: 02-05-2024**

RegisterNo.: 2116231501072 **Name: Kanishka P**

FrequencyofElements

Tofindthefrequencyofnumbersinalistanddisplayinsortedorder.

Constraints:

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

Input:

1687949068145

output:

12

42

51

682

791

901

Forexample:

Input	Result
435345	32 42 52

Program:

```
defcount_frequency(arr):
    frequency = {}

    # Count the frequency of each number in the list
    for num in arr:
```

```
        frequency[num]=frequency.get(num,0)+1

    # Sort the dictionary based on keys
    sorted_frequency=sorted(frequency.items())

    # Print the frequency of each number
    for num, freq in sorted_frequency:
        print(num,freq)

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

# Count the frequency and print the result
count_frequency(arr)
```

Output:

	Input	Expected	Got	
✔	4 3 5 3 4 5	3 2 4 2 5 2	3 2 4 2 5 2	✔
✔	12 4 4 4 2 3 5	2 1 3 1 4 3 5 1 12 1	2 1 3 1 4 3 5 1 12 1	✔
✔	5 4 5 4 6 5 7 3	3 1 4 2 5 3 6 1 7 1	3 1 4 2 5 3 6 1 7 1	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.