

[Dashboard](#) / [My courses](#) / [PSPP/PUP](#) / [Searching techniques: Linear and Binary](#) / [Week10_Coding](#)

Started on	Thursday, 6 June 2024, 4:49 PM
State	Finished
Completed on	Friday, 7 June 2024, 8:07 PM
Time taken	1 day 3 hours
Marks	5.00/5.00
Grade	100.00 out of 100.00

Question **1**

Correct

Mark 1.00 out of 1.00

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

Answer: (penalty regime: 0 %)

```
1 a=input().split()
2 x=list(a)
3 dict={}
4 for element in a:
5     if element in dict:
6         dict[element]+=1
7     else:
8         dict[element]=1
9 s=sorted(dict.items(), key = lambda y:int(y[0]))
10 for key,value in s:
11     print(f"{key} {value}")
```

	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.

Question 2

Correct

Mark 1.00 out of 1.00

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

$A[i-1] \leq A[i] \geq A[i+1]$ for middle elements. $[0 < i < n-1]$

$A[i-1] \leq A[i]$ for last element $[i=n-1]$

$A[i] \geq A[i+1]$ 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

Answer: (penalty regime: 0 %)

```

1 a=int(input())
2 b=input().split()
3 x=list(map(int,b))
4 y=[]
5 for i in range (len(x)):
6     if (i==0 or x[i]>= x[i-1] and i==len(x)-1 or x[i]>= x[i+1]):
7         y.append(x[i])
8 for i in range (len(y)):
9     print (y[i], end=" ")

```

	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.

Question **3**
Correct
Mark 1.00 out of 1.00

Bubble Sort is the simplest [sorting](#) algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order. You read an [list](#) of numbers. You need to arrange the elements in ascending order and print the result. The [sorting](#) should be done using bubble sort.

Input Format: The first line reads the number of elements in the array. The second line reads the array elements one by one.

Output Format: The output should be a sorted [list](#).

For example:

Input	Result
6 3 4 8 7 1 2	1 2 3 4 7 8
5 4 5 2 3 1	1 2 3 4 5

Answer: (penalty regime: 0 %)

```
1 a=int(input())
2 b=input().split()
3 x=list(b)
4 y=sorted(map(int,x))
5 for i in y:
6     print(i,end=" ")
```

	Input	Expected	Got	
✓	6 3 4 8 7 1 2	1 2 3 4 7 8	1 2 3 4 7 8	✓
✓	6 9 18 1 3 4 6	1 3 4 6 9 18	1 3 4 6 9 18	✓
✓	5 4 5 2 3 1	1 2 3 4 5	1 2 3 4 5	✓

Passed all tests! ✓

Correct
Marks for this submission: 1.00/1.00.

Question **4**
Correct
Mark 1.00 out of 1.00

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:

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

Answer: (penalty regime: 0 %)

```
1 def bubble_sort(arr):
2     num_swaps=0
3     n=len(arr)
4     for i in range (n):
5         swapped= False
6         for j in range (0,n-i-1):
7             if arr[j]>arr[j+1]:
8                 arr[j], arr[j+1]=arr[j+1],arr[j]
9                 num_swaps += 1
10                swapped= True
11            if not swapped:
12                break
13        return num_swaps
14 n=int(input())
15 arr=list(map(int,input().split()))
16 num_swaps=bubble_sort(arr)
17 print("List is sorted in", num_swaps,"swaps.")
18 print("First Element:",arr[0])
```

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

	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.

//

Question 5

Correct

Mark 1.00 out of 1.00

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

Answer: (penalty regime: 0 %)

```
1 x=int(input())
2 y=input().split()
3 a=list(y)
4 a=sorted(a)
5 for i in a:
6     print(i,end=" ")
```

	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 57 70	✓
✓	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.

[◀ Week10_MCQ](#)

Jump to...

Sorting ►