

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

Started on	Monday, 10 June 2024, 10:23 PM
State	Finished
Completed on	Monday, 10 June 2024, 10:34 PM
Time taken	10 mins 10 secs
Marks	4.00/5.00
Grade	80.00 out of 100.00

Question 1

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

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

$A[i-1] \leq A[i]$ 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

Answer: (penalty regime: 0 %)

```

1 n = int(input())
2 A = list(map(int, input().split()))
3 peak_elements = []
4 for i in range(n):
5     if (i == 0 and A[i] >= A[i+1]) or (i == n-1 and A[i] >= A[i-1]) or (0 < i < n-1 and A[i-1] <= A[i] >= A[i+1]):
6         peak_elements.append(A[i])
7 print(" ".join(map(str, peak_elements)))

```

Input	Expected	Got	
-------	----------	-----	--

	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 **2**

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 def bubble_sort(arr):
2     n = len(arr)
3     for i in range(n):
4         swapped = False
5         for j in range(0, n-i-1):
6             if arr[j] > arr[j+1]:
7                 arr[j], arr[j+1] = arr[j+1], arr[j]
8                 swapped = True
9         if not swapped:
10            break
11    return arr
12    n = int(input())
13    arr = list(map(int, input().split()))
14    sorted_arr = bubble_sort(arr)
15    print(" ".join(map(str, sorted_arr)))

```

	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 3

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 a=int(input())
2 b = input().split()
3 b = [int(i) for i in b]
4 b.sort()
5 for i in b:
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.

Question 4

Correct

Mark 1.00 out of 1.00

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

Answer: (penalty regime: 0 %)

```

1 def binary_search(arr,x):
2     arr.sort()
3     left,right=0,len(arr)-1
4     while left <=right:
5         mid=(left+right)//2
6         if arr[mid]==x:
7             return True
8         elif arr[mid]<x:
9             left=mid+1
10        else:
11            right=mid-1
12
13    return False
14
15 numbers=list(map(int,input().split(',')))
16 target=int(input())
17 result=binary_search(numbers,target)
18 print(result)

```

	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.

Question **5**

Not answered

Mark 0.00 out of 1.00

To find the frequency of numbers in a [list](#) and display in sorted order.

Constraints:

$1 \leq n$, $\text{arr}[i] \leq 100$

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 ||

[◀ Week10_MCQ](#)

Jump to...

Sorting ▶