

GE19211 / GE23233 / GE23231 - PSPP/PUP

Dashboard / My courses / PSPP/PUP / Searching techniques: Linear and Binary / Week10_Coding

Quiz navigation



Show one page at a time

Finish review

Started on	Saturday, 25 May 2024, 8:31 AM
State	Finished
Completed on	Saturday, 25 May 2024, 8:53 AM
Time taken	22 mins 15 secs
Marks	5.00/5.00
Grade	100.00 out of 100.00

Question **1**

Correct

Mark 1.00 out of 1.00

🚩 Flag question

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
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 def frequency_counter(arr):
2     frequency_dict = {}
3     for num in arr:
4         if num in frequency_dict:
5             frequency_dict[num] += 1
6         else:
7             frequency_dict[num] = 1
8     sorted_frequency = sorted(frequency_dict.items())
9     for num, freq in sorted_frequency:
10        print(num, freq)
11 arr = list(map(int, input().split()))
12 frequency_counter(arr)
13
14
15
```

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

🚩 Flag question

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	1 2 3 4 7 8
3 4 8 7 1 2	
5	1 2 3 4 5
4 5 2 3 1	

Answer: (penalty regime: 0 %)

```
1 def bubble_sort(arr):
2     n = len(arr)
3     for i in range(n):
4         for j in range(0, n-i-1):
5             if arr[j] > arr[j+1]:
6                 arr[j], arr[j+1] = arr[j+1], arr[j]
7     n = int(input())
8     arr = list(map(int, input().split()))
9     bubble_sort(arr)
10    print(arr)
11
```

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

Passed all tests! ✓

Correct
Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

🚩 Flag question

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:

Input	Result
5	Yes
8 9 12 15 3	
11	
6	No
2 9 21 32 43 43 1	
4	

Answer: (penalty regime: 0 %)

```
1 def has_sum_to_k(arr, k):
2     seen = set()
3     for num in arr:
4         complement = k - num
5         if complement in seen:
6             return "Yes"
7         seen.add(num)
8     return "No"
9 n = int(input())
10 arr = list(map(int, input().split()))
11 k = int(input())
12 print(has_sum_to_k(arr, k))
13
```

	Input	Expected	Got	
✓	5	Yes	Yes	✓
	8 9 12 15 3			
	11			
✓	6	No	No	✓
	2 9 21 32 43 43 1			
	4			
✓	6	Yes	Yes	✓
	13 42 31 4 8 9			
	17			

Passed all tests! ✓

Correct
Marks for this submission: 1.00/1.00.

Question **4**

Correct

Mark 1.00 out of 1.00

🚩 Flag question

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

For example:

Input	Result
5	3 4 5 6 8
6 5 4 3 8	

Answer: (penalty regime: 0 %)

```
1 def merge_sort(arr):
2     if len(arr) > 1:
3         mid = len(arr) // 2
4         left_half = arr[:mid]
5         right_half = arr[mid:]
6
7         merge_sort(left_half)
8         merge_sort(right_half)
9
10        i = j = k = 0
11        while i < len(left_half) and j < len(right_half):
12            if left_half[i] < right_half[j]:
13                arr[k] = left_half[i]
14                i += 1
15            else:
16                arr[k] = right_half[j]
17                j += 1
18            k += 1
19        while i < len(left_half):
20            arr[k] = left_half[i]
21            i += 1
22            k += 1
23        while j < len(right_half):
24            arr[k] = right_half[j]
25            j += 1
26            k += 1
27    n = int(input())
28    arr = list(map(int, input().split()))
29    merge_sort(arr)
30    print(arr)
31
32
```

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

Passed all tests! ✓

Correct
Marks for this submission: 1.00/1.00.

Question **5**

Correct

Mark 1.00 out of 1.00

🚩 Flag question

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:

Input	Result
4	12 8
12 3 6 8	

Answer: (penalty regime: 0 %)

```
1 def find_peak(arr):
2     peak_elements = []
3     if arr[0] >= arr[1]:
4         peak_elements.append(arr[0])
5     for i in range(1, len(arr) - 1):
6         if arr[i - 1] <= arr[i] >= arr[i + 1]:
7             peak_elements.append(arr[i])
8     if arr[-1] >= arr[-2]:
9         peak_elements.append(arr[-1])
10
11    return peak_elements
12 n = int(input())
13 arr = list(map(int, input().split()))
14 peak_elements = find_peak(arr)
15 print("peak_elements")
16
```

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

Passed all tests! ✓

Correct
Marks for this submission: 1.00/1.00.

Finish review

← Week10_MCQ

Jump to... 🔍

Sorting ➡

✉ Contact site support

You are logged in as ANJALI G S 2022-BIOMED-A (Log out)

PSPP/PUP

Data retention summary