▲ HEMADARSHINI R S 2022-BIOMED-A H2 ~



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

Quiz navigation Started on Sunday, 26 May 2024, 8:00 PM State Finished Show one page at a time Marks 5.00/5.00 Finish review Question 1 Correct Mark 1.00 out of 1.00 Input Flag question

REC-PS

Completed on Sunday, 26 May 2024, 8:09 PM Time taken 8 mins 12 secs Grade 100.00 out of 100.00 Write a Python program to sort a list of elements using the merge sort algorithm. For example: Result 3 4 5 6 8 6 5 4 3 8

Answer: (penalty regime: 0 %) 1 | n = int(input()) 2 arr = list(map(int, input().split())) 3 * def merge_sort(arr): if len(arr) <= 1: 5 return arr mid = len(arr) // 2left_half = merge_sort(arr[:mid]) right_half = merge_sort(arr[mid:]) return sorted(left_half + right_half) print(*merge_sort(arr)) 10 11 12 13

Expected Got Input

3 4 5 6 8

23 43 49 86

3 4 5 6 8

23 43 49 86

14 21 27 41 43 45 46 57 70 14 21 27 41 43 45 46 57 70

86 43 23 49 Passed all tests! < Correct Marks for this submission: 1.00/1.00. Given an listof integers, sort the array in ascending order using the Bubble Sort algorithm above. Once sorted, print the following three lines: List is sorted in numSwaps swaps., where numSwaps is the number of swaps that took place.

Question 2

Mark 1.00 out of

Flag question

Correct

First Element: firstElement, the first element in the sorted list. 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

6 5 4 3 8

14 46 43 27 57 41 45 21 70

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

 $1 <= a[i] <= 2x10^6$. **Output Format** You must print the following three lines of output: List is sorted in numSwaps swaps., where numSwaps is the number of swaps that took place. First Element: firstElement, the first element in the sorted list. 3. Last Element: lastElement, the *last* element in the sorted list. Sample Input 0

123 First Element: 1 Last Element: 3

22 n = int(input()) 23 a = list(map(int, input().split())) 24 25 26 sorted_list, num_swaps = bubble_sort(a) 27 28 29 print(f"List is sorted in {num_swaps} swaps.") 31 | print(f"First Element: {sorted_list[0]}") 32 print(f"Last Element: {sorted_list[-1]}") 33 Input 3 2 1 Passed all tests! <

Constraints:

Input:

output:

12

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

1 68 79 4 90 68 1 4 5

11

12 13

14 15 16

17 18

19 20

21

Question 3

Mark 1.00 out of

F Flag question

Correct

2 6 8 9 + 10 arr = list(map(int, input().split())) 11 12 frequency_counter(arr)

Input

4 3 5 3 4 5

12 4 4 4 2 3 5

5 4 5 4 6 5 7 3 3 1

Result

False

3,5,9,45,42 True

print(C)

Input

1,2,3,5,8

√ 3,5,9,45,42

✓ 52,45,89,43,11 True

Marks for this submission: 1.00/1.00.

Result

peaks = []

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

peaks.append(A[0]) for i in range(1, n - 1):

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

peaks.append(A[i])

12 8

The first line contains a single integer n, the length of A.

6

42

11

Correct

Input Format

Passed all tests! <

42

Passed all tests! < Correct Marks for this submission: 1.00/1.00. Question 4 Correct For example: Mark 1.00 out of Input Flag question 1,2,3,5,8

Question 5 Correct Mark 1.00 out of 1.00 F Flag question

The second line contains n space-separated integers, A[i]. **Output Format** Print peak numbers separated by space. Sample Input 5 891026 Sample Output 10 6 For example: Input

12 3 6 8 Answer: (penalty regime: 0 %) 1 - def find_peak_elements(n, A): 2 3 + 6 + 8 9 10 11 12

if n > 1 and A[-1] >= A[-2]: peaks.append(A[-1]) return peaks n = int(input()) A = list(map(int, input().split())) peak_elements = find_peak_elements(n, A) 13 print(*peak_elements) 14 15 16 **Expected Got** Input 15 10 9 6 15 10 9 6 🗸 15 7 10 8 9 4 6 12 8

Passed all tests! < Correct Marks for this submission: 1.00/1.00. Jump to...

12 8 12 3 6 8 Sorting -

Sample Output 0 List is sorted in 0 swaps. For example: Result Input List is sorted in 3 swaps. 3 2 1 First Element: 1 Last Element: 3 List is sorted in 4 swaps. 1 9 2 8 4 First Element: 1 Last Element: 9 Answer: (penalty regime: 0 %) 1 + def bubble_sort(arr): n = len(arr)3 num_swaps = 0 4 for i in range(n): swapped = False 8 for j in range(0, n - i - 1): 9 if arr[j] > arr[j + 1]: 10

Expected Got List is sorted in 3 swaps. List is sorted in 3 swaps. 🗸 First Element: 1 First Element: 1 Last Element: 3 Last Element: 3 List is sorted in 4 swaps. List is sorted in 4 swaps. 🗸 1 9 2 8 4 First Element: 1 First Element: 1 Last Element: 9 Last Element: 9 Marks for this submission: 1.00/1.00. To find the frequency of numbers in a list and display in sorted order.

arr[j], arr[j + 1] = arr[j + 1], arr[j]

num_swaps += 1 swapped = True

if not swapped:

break

return arr, num_swaps

Result 4 2 5 2 Answer: (penalty regime: 0 %) 1 | def frequency_counter(arr): frequency_dict = {} for num in arr: if num in frequency_dict: frequency_dict[num] += 1

frequency_dict[num] = 1

for num, freq in sorted_frequency:

print(num, freq)

3 2

4 2

5 2

4 3

5 1

12 1

7 1

sorted_frequency = sorted(frequency_dict.items())

Answer: (penalty regime: 0 %) 1 | A = sorted(list(map(int, input().split(',')))) 2 B = int(input()) 3 left, right = 0, len(A) - 1 4 C = False 5 while left <= right: mid = (left + right) // 2if A[mid] == B: C = True break elif A[mid] < B: left = mid + 1else:

> right = mid - 1 **Expected Got** False False 🗸 True True

True 🗸 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 $A[i-1] \le A[i] \ge a[i+1]$ for middle elements. $[0 \le i \le n-1]$ $A[i-1] \le A[i]$ for last element [i=n-1]A[i] > = A[i+1] for first element [i=0]

Finish review

■ Week10_MCQ