

[Dashboard](#) / [My courses](#) / [PSPP/PUP](#) / [Searching techniques: Linear and Binary](#) / [Week10 Coding](#)

Started on	Sunday, 26 May 2024, 11:54 PM
State	Finished
Completed on	Monday, 27 May 2024, 12:06 AM
Time taken	12 mins 27 secs
Marks	5.00/5.00
Grade	100.00 out of 100.00

Question 1

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 def merge_sort(arr):
2     if len(arr)<=1:
3         return arr
4     mid = len(arr)//2
5     l = arr[:mid]
6     r = arr[mid:]
7     l = merge_sort(l)
8     r = merge_sort(r)
9     return merge(l,r)
10 def merge(left,right):
11     merged = []
12     li,ri=0,0
13     while li<len(left)and ri<len(right):
14         if left[li]<right[ri]:
15             merged.append(left[li])
16             li+=1
17         else:
18             merged.append(right[ri])
19             ri+=1
20     merged.extend(left[li:])
21     merged.extend(right[ri:])
22     return merged
23 n = int(input())
24 arr = list(map(int, input("").split()))
25 sorted_arr = merge_sort(arr)
26 print(' '.join(map(str,sorted_arr)))
```

	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 **2**

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 = list(map(int, input().split()))
2 for B in sorted(set(A)):
3     print(B, A.count(B))
```

	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	✓

	Input	Expected	Got	
✓	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 **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 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(' '.join(map(str,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 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

- $2 \leq n \leq 600$
- $1 \leq a[i] \leq 2 \times 10^6$.

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     n = len(arr)
3     num_swaps = 0
4
5     for i in range(n):
6
7         swapped = False
8
9         for j in range(0, n - i - 1):
10             if arr[j] > arr[j + 1]:
11

```

```
12         arr[j], arr[j + 1] = arr[j + 1], arr[j]
13         num_swaps += 1
14         swapped = True
15
16
17     if not swapped:
18         break
19
20     return arr, num_swaps
21
22
23 n = int(input())
24 a = list(map(int, input().split()))
25
26
27 sorted_list, num_swaps = bubble_sort(a)
28
29
30 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]}")
```

	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

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

Answer: (penalty regime: 0 %)

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


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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

◀ [Week10_MCQ](#)

Jump to...

Sorting ▶