

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

Started on	Monday, 10 June 2024, 10:50 PM
State	Finished
Completed on	Monday, 10 June 2024, 10:58 PM
Time taken	8 mins 17 secs
Marks	4.00/5.00
Grade	80.00 out of 100.00

## Question 1

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 x(a,b):
2     a.sort()
3     l,r=0,len(a)-1
4     while l<=r:
5         m=(l+r)//2
6         if a[m]==b:
7             return True
8         elif a[m]<b:
9             l=m+1
10        else:
11            r=m-1
12    return False
13 n=list(map(int,input().split(','))
14 t=int(input())
15 result=x(n,t)
16 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 **2**

Correct

Mark 1.00 out of 1.00

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

**Constraints:**
 $1 \leq n$ ,  $\text{arr}[i] \leq 100$ 
**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 n=list(map(int,input().split()))
2 f={}
3 for num in n:
4     f[num]=f.get(num,0)+1
5 sf=sorted(f.items())
6 for num,freq in sf:
7     print(num,freq)

```

	Input	Expected	Got	
✓	4 3 5 3 4 5	3 2 4 2 5 2	3 2 4 2 5 2	✓

	Input	Expected	Got	
✓	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 3

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 n=int(input())
2 numbers=list(map(int,input().split()))
3 k=int(input())
4 for i in range(n):
5     for j in range(i+1,n):
6         if numbers[i]+numbers[j]==k:
7             print('Yes')
8             exit()
9 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.

#### Question 4

Not answered

Mark 0.00 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 ||

## Question 5

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

```

	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.

◀ [Week10\\_MCQ](#)

Jump to...

Sorting ▶