Week-10

1)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 3 6 8 | 12 8 |

Program:

def findPeaks(n, A):

peaks = []

if n > 0 and (n == 1 or A[0] >= A[1]):

peaks.append(A[0])

for i in range(1, n-1):

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

peaks.append(A[i])

if n > 1 and A[n-1] >= A[n-2]:

peaks.append(A[n-1])

print(" ".join(map(str, peaks)))

n = int(input())

A = list(map(int, input().split()))

findPeaks(n, A)

output:

InputExpectedGot

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

2) 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 |

Program:

a = list(map(int, input().split(',')))

x = int(input())

a.sort()

l = 0

h = len(a) - 1

while l <= h:

m = (h + l) // 2

if a[m] < x:

l = m + 1

elif a[m] > x:

h = m - 1

else:

print(True)

break

else:

print(False)

output:

| **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 |  |

3)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 |

Program:

arr = list(map(int, input().split()))

frequency\_dict = {}

for num in arr:

if num in frequency\_dict:

frequency\_dict[num] += 1

else:

frequency\_dict[num] = 1

sorted\_frequency = sorted(frequency\_dict.items())

for key, value in sorted\_frequency:

print(key, value)

output:

|  | **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 |  |
|  | 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 |  |

4)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 |

Program:

neil=int(input())

a = list(map(int, input().split()))

key=int(input())

fg=0

for i in range(neil):

for j in range(0,neil):

if(a[i]!=a[j]):

if(a[i]+a[j]==key):

fg+=1

if(fg==0):

print("No")

else:

print("Yes")

output:

InputExpectedGot

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

5)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 |

Program:

def merge\_sort(arr):

if len(arr) > 1:

mid = len(arr) // 2

left\_half = arr[:mid]

right\_half = arr[mid:]

merge\_sort(left\_half)

merge\_sort(right\_half)

i = j = k = 0

while i < len(left\_half) and j < len(right\_half):

if left\_half[i] < right\_half[j]:

arr[k] = left\_half[i]

i += 1

else:

arr[k] = right\_half[j]

j += 1

k += 1

while i < len(left\_half):

arr[k] = left\_half[i]

i += 1

k += 1

while j < len(right\_half):

arr[k] = right\_half[j]

j += 1

k += 1

def print\_list(arr):

for num in arr:

print(num, end=" ")

n = int(input())

arr = list(map(int, input().split()))

merge\_sort(arr)

print\_list(arr)

output:

| **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 |  |