10.Searching technique: linear and binary

**Ex. No. : 10.1 Date:06/06/2024**

**Register No.: 231801047 Name: HANNAH JAMES**

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 m\_s(a):

if len(a) > 1:

m = len(a) // 2

l\_h = arr[:m]

r\_h = arr[m:]

m\_s(l\_h)

m\_s(r\_h)

i = j = k = 0

while i < len(l\_h) and j < len(r\_h):

if l\_h[i] < r\_h[j]:

a[k] = l\_h[i]

i += 1

else:

a[k] = r\_h[j]

j += 1

k += 1

while i < len(l\_h):

a[k] = l\_h[i]

i += 1

k += 1

while j < len(r\_h):

a[k] = r\_h[j]

j += 1

k += 1

def main():

n = int(input())

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

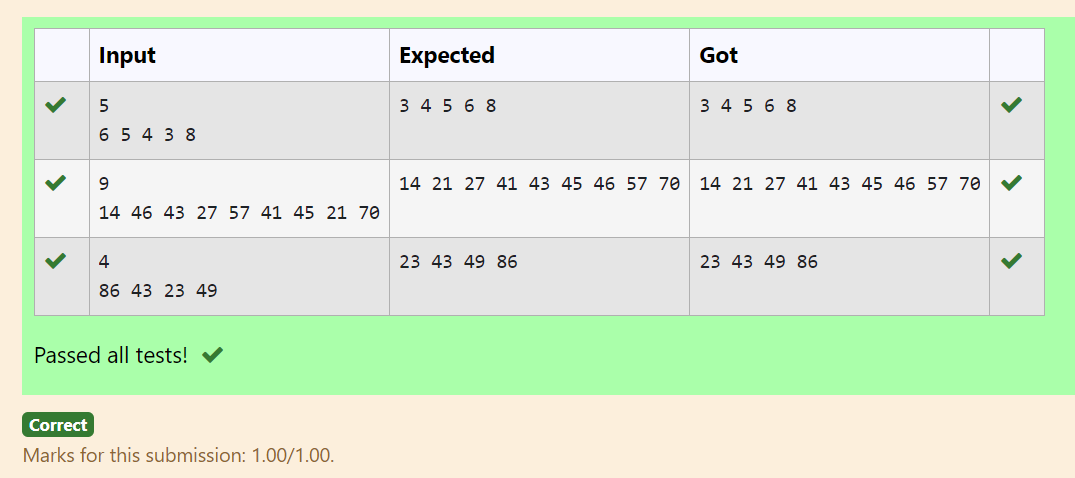
m\_s(a)

for num in a:

print(num, end=" ")

if \_\_name\_\_ == "\_\_main\_\_":

main()



**Ex. No. : 10.2 Date: 06/06/2024**

**Register No.: 231801047 Name: HANNAH JAMES**

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:

def c\_f(a):

f = {}

for num in a:

if num in f:

f [num] += 1

else:

f [num] = 1

s\_f = sorted(f.items())

return s\_f

def main():

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

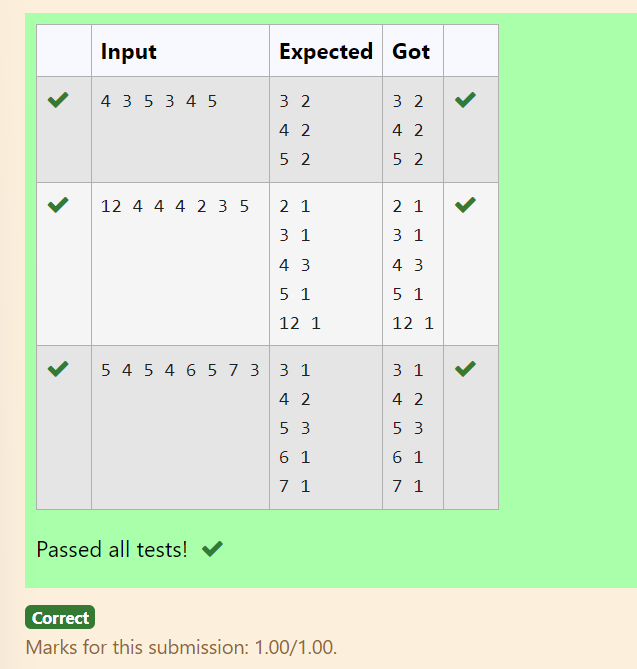
s\_f = c\_f (a)

for num, freq in s\_f:

print(num, freq)

if \_\_name\_\_ == "\_\_main\_\_":

main()



**Ex. No. : 10.3 Date: 06/06/2024**

**Register No.: 231801047 Name: HANNAH JAMES**

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 |

PROGRAM:

def b\_s (a):

n = len(a)

for i in range(n):

for j in range(0, n-i-1):

if a [j] > a [j+1]:

a [j], a [j+1] = a [j+1], a [j]

def main():

n = int(input())

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

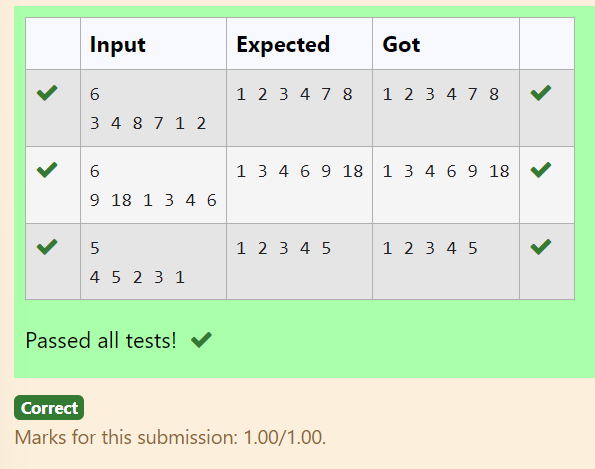
b\_s(a)

for num in a:

print(num, end=" ")

if \_\_name\_\_ == "\_\_main\_\_":

main()



**Ex. No. : 10.4 Date: 06/06/2024**

**Register No.: 231801047 Name: HANNAH JAMES**

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:

n=int(input())

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

k=int(input())

fg=0

for i in range(neil):

for j in range(0,nl):

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

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

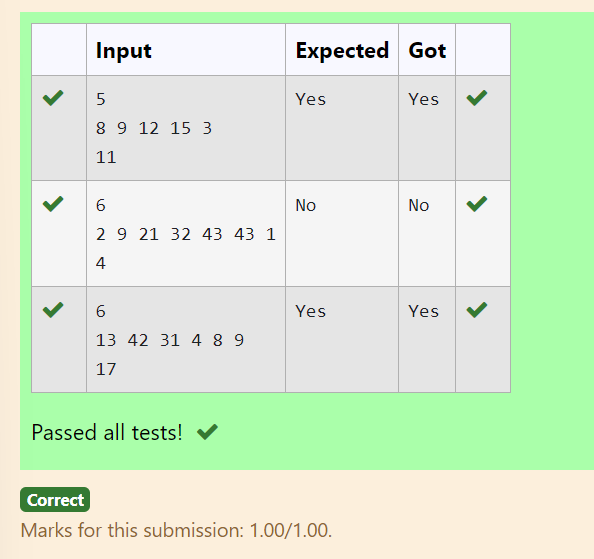
fg+=1

if(fg==0):

print("No")

else:

print("Yes")



**Ex. No. : 10.5 Date: 06/06/2024**

**Register No.: 231801047 Name: HANNAH JAMES**

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:

def b\_s (a, x):

a.sort()

l, r = 0, len(a) - 1

while l <= r:

m = (l+ r) // 2

if a [m] == x:

return True

elif a [m] < x:

l = m + 1

else:

r = m - 1

return False

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

t= int(input())

r = b\_s (n, t)

print(r)

