10 - Searching & Sorting

Ex. No. : 10.1 **Date: 01.06.24**

Register No.: 230701385 **Name S. Vishwak**

.

Bubble Sort

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 bubble_sort(arr):
```

```
    n=len(arr)
    for i in range(n-1):
        for j in range(0,n-i-1):
            if arr[j]>arr[j+1]:
                arr[j],arr[j+1]=arr[j+1],arr[j]
    return arr
n=int(input())
arr=list(map(int,input().split()))
result=[]
result=bubble_sort(arr)
print(*result)
```

	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	✔

Ex. No. : 10.2 **Date: 01.06.24**

Register No.: 230701385 **Name S. Vishwak**

.

Peak Element

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 findPeak(arr, n) :
    r=[]
    if(arr[0]>=arr[1]):
        r.append(arr[0])
    for i in range(1,n-1):
        if(arr[i]>=arr[i-1] and arr[i]>=arr[i+1]):
            r.append(arr[i])
    if(arr[n-1]>=arr[n-2]):
        r.append(arr[n-1])
    return r


n=int(input())
arr=list(map(int,input().split()))
r=findPeak(arr,n)
print(*r)
```

	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	✔

Ex. No. : 10.3 **Date: 01.06.24**

Register No.: 230701385 **Name: S. Vishwak**

.

Merge Sort

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
        L=arr[:mid]
        R=arr[mid:]
        merge_sort(L)
        merge_sort(R)
        i=j=k=0
        while i < len(L) and j <len(R):
            if L[i]<R[j]:
                arr[k]=L[i]
                i+=1
            else:
                arr[k]=R[j]
                j+=1
            k+=1
```

```
        while i <len(L):
            arr[k]=L[i]
            i+=1
            k+=1
        while j<len(R):
            arr[k]=R[j]
            j+=1
            k+=1
n=int(input())
arr=list(map(int,input().split()))
merge_sort(arr)
print(*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	✔

Ex. No. : 10.4 **Date: 01.06.24**

Register No.: 230701385 **Name: S. Vishwak**

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 bin_search(arr,x):
    arr.sort()
    l,r=0,len(arr)-1
    while l<=r:
        mid=(l+r)//2
        if(arr[mid]==x):
            return True
        elif arr[mid]<x:
            l=mid+1
        else:
            r=mid-1
    return False
num=list(map(int,input().split(',')))
```

```
key=int(input())
result=bin_search(num,key)
print(result)
```

	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	✔

Ex. No. : 10.5 **Date: 01.06.24**

Register No.: 230701385 **Name: S. Vishwak**

.

Frequency of Elements

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

Input	Result
	5 2

Program:

```
numbers=list(map(int,input().split()))
freq={}
for n in numbers:
    freq[n]=freq.get(n,0)+1
sorted_freq=sorted(freq.items())
for num,freq in sorted_freq:
    print(num,freq)
```

	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	✔