10 - Searching & Sorting

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

Register No.: **Name:**

Bubble Sort

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
```

```
def bubble_sort(arr):
    n = len(arr)
    num_swaps = 0

    for i in range(n):
        swapped = False
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
                num_swaps += 1
                swapped = True
        if not swapped:
            break
```

```
    print(f"List is sorted in {num_swaps} swaps.")
    print(f"First Element: {arr[0]}")
    print(f"Last Element: {arr[-1]}")


n = int(input())
a = list(map(int, input().split()))
bubble_sort(a)
```

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

Register No.: **Name:**

Merge Sort

Write a Python program to sort a list of elements using the merge sort algorithm.

```
n = int(input())
array = input().split()
for i in range(n):
    array[i] = int(array[i])
for i in range(n):
    swapped = False
    for j in range(0, n - i - 1):
        if array[j] > array[j + 1]:
            array[j], array[j + 1] = array[j + 1], array[j]
            swapped = True
    if not swapped:
        break

for i in range(n):
    print(array[i], end=' ')
print()
```

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

Register No.: **Name:**

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 clement [i=0]

Ex. No. : 10.5 Date:

Register No.: Name:

Frequency of Elements

To find the frequency of numbers in a list and display in sorted order.

Constraints:

1<=n, arr[i]<=100

```
input_numbers = input().strip().split()
numbers = [int(x) for x in input_numbers]
frequency = {}
for number in numbers:
if number in frequency:
frequency[number] += 1
else:
frequency[number] = 1
sorted_numbers = sorted(frequency.keys())
for number in sorted_numbers:
print(number, frequency[number])
```

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

Register No.: **Name:**

Binary Search

Write a Python program for binary search.

```
def binary_search(arr, x):
    left = 0
    right = len(arr) - 1
    while left <= right:
        mid = left + (right - left) // 2
        if arr[mid] == x:
            return True
        elif arr[mid] < x:
            left = mid + 1
        else:
            right = mid - 1
    return False
def main():
    arr = list(map(int, input().strip().split(',')))
    x = int(input().strip())
    result = binary_search(sorted(arr), x)
    print(result)
main()
```