# 10 - Searching & Sorting

##### For example:

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

178

|  |  |
| --- | --- |
| **Input** | **Result** |
| 5  6 5 4 3 8 | 3 4 5 6 8 |

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

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

179

**Register No.: Name:**

### Merge Sort

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

n = int(input())

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

size = len(arr)

if size > 1:

mid = size // 2

left\_half = arr[:mid]

right\_half = arr[mid:]

left\_size = len(left\_half)

right\_size = len(right\_half)

left\_sorted = []

right\_sorted = []

if left\_size > 1:

left\_mid = left\_size // 2

left\_half[:left\_mid] = sorted(left\_half[:left\_mid])

left\_half[left\_mid:] = sorted(left\_half[left\_mid:])

l, r, k = 0, 0, 0

while l < left\_mid and r < left\_size - left\_mid:

if left\_half[l] < left\_half[left\_mid + r]:

left\_sorted.append(left\_half[l])

l += 1

else:

left\_sorted.append(left\_half[left\_mid + r])

r += 1

k += 1

while l < left\_mid:

left\_sorted.append(left\_half[l])

l += 1

while r < left\_size - left\_mid:

left\_sorted.append(left\_half[left\_mid + r])

r += 1

else:

left\_sorted = sorted(left\_half)

if right\_size > 1:

right\_mid = right\_size // 2

right\_half[:right\_mid] = sorted(right\_half[:right\_mid])

right\_half[right\_mid:] = sorted(right\_half[right\_mid:])

l, r, k = 0, 0, 0

while l < right\_mid and r < right\_size - right\_mid:

if right\_half[l] < right\_half[right\_mid + r]:

right\_sorted.append(right\_half[l])

l += 1

else:

right\_sorted.append(right\_half[right\_mid + r])

r += 1

k += 1

while l < right\_mid:

right\_sorted.append(right\_half[l])

l += 1

while r < right\_size - right\_mid:

right\_sorted.append(right\_half[right\_mid + r])

r += 1

else:

right\_sorted = sorted(right\_half)

i, j, k = 0, 0, 0

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

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

arr[k] = left\_sorted[i]

i += 1

else:

arr[k] = right\_sorted[j]

j += 1

k += 1

while i < len(left\_sorted):

arr[k] = left\_sorted[i]

i += 1

k += 1

while j < len(right\_sorted):

arr[k] = right\_sorted[j]

j += 1

k += 1

for i in arr:

print(i,end=' ')

##### Input Format

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

180

The first line contains an integer,n , the size of the [list](http://118.185.187.137/moodle/mod/resource/view.php?id=1068) a . The second line contains n, space-separated integers a[i].

##### Constraints

· 2<=n<=600

· 1<=a[i]<=2x106.

##### Output Format

You must print the following three lines of output:

1. [List](http://118.185.187.137/moodle/mod/resource/view.php?id=1068) 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](http://118.185.187.137/moodle/mod/resource/view.php?id=1068).
3. Last Element: lastElement, the *last* element in the sorted [list](http://118.185.187.137/moodle/mod/resource/view.php?id=1068).

##### Sample Input 0

3

1 2 3

##### Sample Output 0

[List](http://118.185.187.137/moodle/mod/resource/view.php?id=1068) is sorted in 0 swaps. First Element: 1

Last Element: 3

##### For example:

|  |  |
| --- | --- |
| **Input** | **Result** |
| 3  3 2 1 | List is sorted in 3 swaps. First Element: 1  Last Element: 3 |
| 5  1 9 2 8 4 | List is sorted in 4 swaps. First Element: 1  Last Element: 9 |

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

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

181

**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](http://118.185.187.137/moodle/mod/resource/view.php?id=1068) 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](http://118.185.187.137/moodle/mod/resource/view.php?id=1068).
3. Last Element: lastElement, the *last* element in the sorted [list](http://118.185.187.137/moodle/mod/resource/view.php?id=1068).

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

num\_elements = int(input())

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

n = len(array)

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 array:

print(i,end=' ')

##### Input Format

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

182

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 |

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

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

183

**Register No.: Name:**

### Peak Element

Given an [list](http://118.185.187.137/moodle/mod/resource/view.php?id=1068), 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]

n = int(input())

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

peaks = []

if n == 1:

peaks.append(A[0])

else:

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

peaks.append(A[n - 1])

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

##### For example:

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

184

|  |  |
| --- | --- |
| **Input** | **Result** |
| 1 2 3 5 8  6 | False |
| 3 5 9 45 42  42 | True |

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

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

185

**Register No.: Name:**

### Binary Search

Write a Python program for binary search.

a=input().split(',')

b=input()

if b in a:

print("True")

else:

print("False")

##### Input:

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

186

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 |

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

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

187

**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

n = int(input())

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

k = int(input())

found = False

num\_set = set()

for num in lst:

if k - num in num\_set:

found = True

break

num\_set.add(num)

if found:

print("Yes")

else:

print("No")