CSE221
Lab Assignment 02
Summer 2023

Submission Guidelines:

1. You can code all of them either in Python, CPP, or Java. But you should choose a specific language for all tasks. 2. For each task write separate python files like task1.py, task2.py, and so on.
3. For each problem, take input from files called **"inputX.txt"** and output at **"outputX.txt"**, where X is the task number. 4. Add a hand written explanation of 3-4 lines for each of your solutions in a separate document. You may compile all of your explanations in a single file.
5. Finally zip all the files and rename this zip file as per this format:**LabSectionNo_ID_CSE221LabAssignmentNo_Summer2023.zip** [Example:**LabSection01_21101XXX_CSE221LabAssignment02_Summer2023.zip**]
6. Don't copy from your friends.
7. You MUST follow all the guidelines, naming/file/zipping convention stated above.
   Failure to follow instructions will result in a straight 50% mark deduction.

Task 01: [Points: 15]

Your little brother, Alice, is very fond of playing with integers. One day, Alice was given a list of **N** integers by his school teacher. Now, your brother wants to play a game with you.

Alice will give you an integer, **S**. You have to find if it is possible to find two values from the list (at distinct positions) whose sum is equal to S.

Now you are feeling very tired. So you decided to write a code,

so that it can give you the answer very quickly.

1) Can you write an $O(n^2)$ Solution to solve the problem?
   [Points 5]
2) Come up with an O(n) or O(nlogn) solution. **[Points 10]**

Input

The first line contains two integers N and S (1 <= N <= 10^5, 1 <= S <= 10^9), denoting the length of the list, and the target Sum.

In the next line, there will be N integers $a_1, a_2, ..., a_N$ (1 ≤ a_i ≤ 10^9) separated by space.

Output

Print two integers: **the positions** of the values [**1 based indexing**]. If there are several solutions, you may print any of them. If there are no solutions, print "IMPOSSIBLE".

Sample Input 1 Sample Output 1
4 10   1 2 3 7 1 5
Sample Input 2 Sample Output 2

6 18
    1 5
9 10 1 5 9 8
 	
    [2 6 is also a valid answer]
    [print only one output]

Sample Input 3 Sample Output 3

4 7   IMPOSSIBLE
2 4 6 8

Sample Input 4 Sample Output 4
                    IMPOSSIBLE
3 12 6 1 2

Task 02: [Points: 15]

Alice and Bob are two friends. Alice has a sorted list in ascending order of length **N**. On the other hand, Bob has a sorted list of length **M**. Now, they want to make a sorted list of **N+M** length in ascending order. However, they are not very good at algorithms. Hence, they asked for your help.

Since you are a computer science student, your task is to come up with an efficient algorithm.

1) Find a solution which runs in O(nlogn). **[Points 5]** 2) Come up with a solution which runs in O(n). **[Points 10]**

Input

The first line contains an integer N (1 <= N <= 10^5), denoting the length of Alice's sorted list. In the next line, there will be N integers separated by space.

The third line contains another integer M (1 <= M <= 10^5), denoting the length of Bob's sorted list. In the next line, there will be M integers separated by space.

All the numbers given in the input will fit in a 32-bit signed integer.

It is guaranteed that the given lists will be in sorted order.

Output:

You have to make a sorted list in ascending order from the given lists in ascending order and show the output.

Sample Input/Output:

Sample Input 1	Sample Output 1
4	1 2 2 3 4 5 7 8
1 3 5 7	
4	
2 2 4 8	

Sample Input 2	Sample Output 2
3 2 10 12 6 3 4 6 7 8 9	2 3 4 6 7 8 9 10 12

Sample Input 3	Sample Output 3
4 1 2 3 4 1 10	1 2 3 4 10

Sample Input 4	Sample Output 4
7 2 3 8 8 10 12 14 9 1 1 4 5 6 8 13 15 16	1 1 2 3 4 5 6 8 8 8 10 12 13 14 15 16

Task 03: [Points 10]

In this problem, you will be given a list of numbers. You have to sort the list using the Merge Sort algorithm.

Pseudocode of Merge Sort Algorithm:

```python
def merge(a, b):
        # write your code here
        # Here a and b are two sorted list
        # merge function will return a sorted list after merging a and b


def mergeSort(arr):
    if len(arr) <= 1:
        return arr
```

```
else:
    mid = len(arr)//2
    a1 = mergeSort(............) # write the parameter
    a2 = mergeSort(............) # write the parameter
        return merge(a1, a2) # complete the merge function above
```

Input

The first line contains an integer N (1 <= N <= 10^5), denoting the length of Alice's sorted list. In the next line, there will be N integers separated by space.

Output:

You have to sort the number using the Merge Sort algorithm and show the sorted list.

Sample Input/Output:

Sample Input 1 Sample Output 1
8   9 5 4 6 1 3 2 9   1 2 3 4 5 6 9 9
Sample Input 2 Sample Output 2
1 10   10
Sample Input 3 Sample Output 3
6   8 1 4 2 1 3   1 1 2 3 4 8
Sample Input 4 Sample Output 4

7
7 6 5 4 3 2 1

Task 04 [10 Points]:

1 2 3 4 5 6 7

Alice and you are playing with a list of **N** non negative integers.
Today you will try to find out the maximum number of the list.
Alice writes the following code to find the maximum number.

```
maxValue = arr[0]
for i in range(1,N):
    if maxValue < arr[i]:
        maxValue = arr[i]
```

Recently you have learned merge sort. Now, you are thinking if
you can use the divide and conquer approach to find out the
maximum from the given list.

Input

The first line contains an integer N (1 <= N <= 10^5), denoting
the length of Alice's sorted list. In the next line, there will
be N integers separated by space.

Output:

You have to find out the maximum value from the list using the
divide and conquer approach.

Sample Input/Output:

Sample Input 1	Sample Output 1
8 1 7 13 4 5 7 13 12	13
Sample Input 2	Sample Output 2

```
7
                                        15
5 15 2 3 10 1 9
```

Sample Input 3 Sample Output 3

```
1                              9
9
```

Sample Input 4 Sample Output 4
```
                  10
6
5 2 3 10 1 9
```
What is the time complexity of your code?