Task 1 explanation:

Sub-task-1: To find $O(N^2)$ solution I have chosen selection sort algorithm method. Whenever sum of any two numbers of the list will be equal to 'S', that's when I am adding 1 based indexing to the output. If any of two numbers summation doesn't match the value 'S', then I am adding "IMPOSSIBLE" to the output.

Sub-task-2: To find O(NlogN) solution I took a nested list where I am storing list values with their actual index. After that I am sorting the nested list based on inner list's first value. Then there is a while loop which will add list's left most value and right most value till left is smaller than right. If there summation gets equal to 'S' then it will add the actual index of the main list in the output.. If any of two numbers summation doesn't match the value 'S', then I am adding "IMPOSSIBLE" to the output.

Task 2 explanation:

Sub-task-1: To find O(NlogN) solution I took three variables a=b=0, f = empty string. Then I run a while loop and an if condition to check smaller value which I will add to the output, this loop will run if a and b smaller than N and M. After doing that if there is still any value left in the input two array then I am adding all the values at the end of the output because they are already sorted so I don't have to check anymore.

Sub-task-2: To find O(N) solution I took three variables a=b=0, f = empty string. Then I run a for loop and here again I am checking using conditional statement to find smaller value which I will add to the output, this loop will run the total length of the inputed list. After that if a equal N or b equal to M then I am adding all the leftover values at the end of the output because they are already sorted so I don't have to check anymore.

Task 3 explanation:
This is the merge sort algorithm. Here I set the if the length of the input list is 1 then it will return the list itself. If not then the list will be divided into two parts then this two parts will recursively call then 'mergeSort' function again. After that it will call another function called 'merge' which will compare values and it will add smaller value first in the temporary list called result. If any value not added yet then it will add those leftover values at the end of the result list. Then result list will be returned where it was being called. This process will go on until the whole list is sorted

Task 4 explanation:
Just like previous task 3 here I am using merge sort technique to find the max value of the list. The process will go on until we compare every values of the list to find out the maximum number of the list. Then I am adding the maximum value to the output.