Task 1a: In this code I used two loops to check the combination for the sum which would match the required ~~outp~~ sum. If it matches then it will ~~fei~~ write the output. Here time complexity is $O(n^2)$.

Task 1b: In this code I took two variable which would check from both sides to find the combination ~~to~~ for the output. Here time complexity is $O(N \log N)$.

Task 2a: In this code I just ~~add the~~ added two sorted array into a big array and used ~~builtin~~ builtin sort() function. Its complexity is $O(N \log N)$

Task 2b: In this code ~~I~~ I used ~~loo~~ merge() function to merge two array. In a loop, I checked for smaller values from two array and added to the new untill I got a big sorted array. Time complexity is $O(N)$.

Task 3: In this code I used merge sort to solve the problem. Here a array is broken untill it has only one value then compare will other and finally added to the new array. It is based on divide and conquize. Time complexity of merge sort is O(logN) and merge() is O(N). Total O(NlogN).

Task 4: It is similare from task 3. I used merge sort to make the array small. After that I only added largest value to the new array. Time complexity is O(logN).