Task 1(a)

At first, read the lines from input and assign them into variables. Then run a for loop to n and also a nested for loop from i+1 to n. Then sum = array[i] + array[j]. Then check if it was the equal to the target number or not. If it was then just write the index+1 of both and break else just continue loop. If the loop ends then it means there are no such combination, so just write Impossible. Here the time complexity is $O(n^2)$.

Task 1(b)

To get the time complexity $O(n)$, we use two pointers system. To write it, write two variables, first one initialize it with zero and the second one initialize it with len of the array - 1. Now write a while loop and then the loop will continue until left one is less then the right one. Then sum two of the values of the array. Now check if it is equal to the target. Or else,

If that sum is less than target then just increase the left one. Else, just decrease the right number. If the loop ended, then print "Impossible".

Task2(a)

To sort in O(nlgn), we can just use sort function.

Task2(b)

To sort in O(n), we have to write a while and it will continue until the first pointers of the first array is less than that array length or the second pointer is less then the length of second array. Now first two conditions are for if both array length isn't equal. Then rest of the two conditions are for which one is less than others and append it

Task 3

At first we have to sort it respect with ending time. Then write a while loop and it will continue until the last element of that time pairs. Then just check if the previous work's ending time is equal or less than the next work's starting time.

Task 4

At first sort that time pairs with respect of ending time. Then we have to check which person has less time difference in terms of ending and starting time and assign that work to that person.