Task 1a

The first input is the number of ~~numbe~~ integers for which we need to define if its odd or even. We initiate a loop, read the files accordingly using proper syntax and that input number, turn it into an integer form and eventually, write it inside another output file.

Task 1b

Like the previous task, we initiate a loop after reading the first line. For every iteration, we read lines, split the ~~s~~ lines and according to the middle element we do our operations.

Task-2

For any sorting algorithm, the best case is when the array is already sorted. Here to use buble sort for the best case, we need to initiate a flag. If for the first iteration the condition of inner loop never fulfills then, we can easily say that the array is already sorted and we don't change the flag. Eventually, if the flag doesn't change, then we can easily return from there. The time complexity will be $O(n)$ as the function returns after iteration for the best case.

Task-3

We can form a dictionary using the marks as keys. We sort the keys in a descending order in a different list. Then, for looping the key-list, we ascend the id list for corresponding key value pair, and iterate over to write our output.

Task-4

A dictionary is made to store the name as keys and splitted strings as values. The keys are ~~stro stor~~ sorted in a different list. Corresponding values are iterated using the sorted key-array. Meanwhile, those splitted element's last values were replaced by actual itegers from string. Using those numbers and index we sort all the values in a descending order in that array. Eventually, we write our output accordingly.