

Task 1(a)

• for task 1(a), firstly I've created a .txt file for input, where I took the values. Then for the first line of the input I made it int and started a loop in a range of that. then for each loop I started reading the lines of the input and then converted into integers, and checked whether it's even or not. Then I wrote ~~the~~ in the output.txt file accordingly.

Task 1(b)

This task is quite similar to task 1.

The only new thing I did here is to strip the lines. We know that if we strip any string it makes a list where the elements are acknowledged after space " ". Then I started loop and added, divided, multiplied accordingly.

Task 2

In this problem I have just made a variable `swp` which ~~checks~~ determines if the list is sorted or not. If we get `sorted = "True"` for the first attempt in the nested loop, we furthermore decline to go on again in the loop. Instead we break the loop. That's how we can get the best results.

Task-3

In this problem, firstly I ~~took the first~~ made a sorted

Task-3

In this problem, firstly I determined minimum index number according to the lowest marks. then I checked if marks are same or not. If same, I took that index also as the minimum index. After running 1 loop (in the nested) I swapped marks and ids in a descending manner.

Task-4

This task is quite similar with task 3.

finally I checked the names, the names with lesser letter gets priority here. If the names of the trains are same then the latest departure gets prioritized.