

Task-01(a)

To check whether the number is even or odd, I checked if the remainder is 0 or not when divided by 2.

Task-01(b)

First of all I accessed the inputs using for loop and split the input lines by space and by this I got the operation signs at index 2 and using if and elif condition checked whether I am getting the desired operation sign or not. Then when the condition is fulfilled performed the arithmetic calculation and write the output.

Task-02

To make the ~~code~~ time complexity of the given code for best case ~~scenario~~ ^{$O(n)$} I have used flag so that, if no swap happens, the flag will remain 0 and it will break and the loop will not continue n times and ^{this is how} we can get a constant term for the loop and make the ~~code~~ time complexity $O(n)$ in the best case scenario.

Task-03

At first I created 2 lists ^{at the first one} which consist of IDs and the second one with marks then I passed both list in the function called Insertion sort to sort the list both lists. Here I used Insertion sort because it takes minimum numbers of swaps while sorting and this sorting algo is efficient and can be $O(n)$ in the best case scenario. After sorting I checked if the marks are same or not, if the marks are same then I checked whether the IDs are ^{in the} sorted order for the equal marks. If the IDs are not in sorted order then again I swapped the IDs and this is how I reached to the desired output.

Task-4

I sliced the input lines and stored train name, departure area and time in a list and made another list to store time in minutes. ~~in a list and~~ To convert the time in minute I used time conversion function. Then using insertion sort I sorted the train-name lexicographically and ~~at~~ in the mean time I swapped the time to minutes.

list so that the order does not get mismatched.
~~After that the finishing the sorting of train name~~
When I was sorting the train names the area
an time in the given format was also sorting according to
After that I checked using loop if $\text{index}[k]$ and
 $\text{index}[kt][0]$ is same or not and the departure
time in minute of the train name is same or not
and the departure time in minute of $\text{index}[k] < \text{index}[kt]$
then according to the condition I again swapped
and ultimately came to the desired output.