Name: Razin Sufian

ID: 22301219

Partner name: Sabid Mahmud

ID: 22301172

Assignment 1 code explanation

CSE 221 ; section : 16

Task 1a:

```
data = inpt.readlines()
for i in rage (total int_number):
    if Even :
        write it in the output file as even
    eles:
        write it in the output file as odd
```

Task 1b:

first build a small calculate function. which will take ~~two~~ three parameters. first two are number, third one is a string ; for the sign.

then after reading the input file data, split it and passing to the function and writing it on the output.

Task 21

The sorted function is a normal buble function; but it has some litt changed for the best case senario; that ~~it won't be~~ the loop would break when the list is became sorted.

first we cued a flag; "sorted = True" we can see in line eleven we are comparing the numbers. if ~~there is~~ 11 number line condition is true flag turns to "false".

But for the best case, when the 11 th line condition will never be true the flag will remain True. So, after in the complitation of that specific traversal ; in 14th line ~~sorted==~~ condition will be true ; then it will be break from the entire loop.

thus it is efficent for the best case scenario.

Task 3:

Here from the hint we understand that we will have to use selection sort.

now; there are 3 steps.

1. sort the marks
2. while sorting replace the index number of the list which contains ID
3. if two numbers are same compare it with ID

So for completing condition ① we use normal selection sort. while

we swap we not only swap the elements of Marks contained list but also swap the ID contained list thus we complete condition 3.

in 13th line we chek if the two numbers are same or not. if same then we changed the index number by comparing with the ID in line 14th.

Task 4

here we actually have to sort the data_list ; which we read from the input file.

if we compare str-1 > str-2 ; python compiler returns true if the first string is lexicographically larger than the second string. otherwise it ~~writer~~ returns false.

So, I used normal selection sort for that ; split each data and compare the [0] index.

if index [0] is same then ; I checked at the 10th line that the time is also same or not.

if time is also same ; then which input comes first will remain first. So ; as I used selection sort I didn't sorted the element if the Name and time both are same. that's why in line 11th I used pass.

if time is not same then at line 16 and 19 I ~~compared~~

splited and compared the time and sorted accordingly.