Programming Assignment 08

-

Files and exceptions

Instructions

This programming assignment consists of **2 programming exercises**.
You have to:

1. **download** the empty Python files on NYU Classes
2. **edit** them according to the assignment
3. **verify** on your computer that it works
4. **upload** them back on NYU Classes (**do not change the filenames**)

Exercise 1 - File merging

Suppose we have **two text files** `file1.txt` and `file2.txt` in the same folder as your python program (you can download the sample files on NYU Classes and put it in the same folder as your program, or create your own text files).
Write a program (in the `main()` in `exercise1.py`) which inserts all the content from `file2.txt` to the beginning of `file1.txt`.
After executing your program, only `file1.txt` will be updated, `file2.txt` will not change.

Sample example 1:

- before execution of the program

	file1.txt		file2.txt
1	ICP	1	Coding
2	Summer	2	Assignment
3	2021	3	08
4		4	for
		5	
str	'ICP\nSummer\n2021\n'	str	'Coding\nAssignment\n08\nfor\n'

- after execution of the program

	file1.txt		file2.txt
1	Coding	1	Coding
2	Assignment	2	Assignment
3	08	3	08
4	for	4	for
5	ICP	5	
6	Summer		
7	2021		
8			

Sample example 2 (notice that the trailing `\n` were important in previous sample):

- before execution of the program

	file1.txt		file2.txt
1	1 2 3	1	4 5 6
str	'1 2 3'	str	'4 5 6'

- after execution of the program

	file1.txt		file2.txt
1	4 5 61 2 3	1	4 5 6

Exercise 2 - I won't let it crash!

Write a program (in the `main()` in `exercise2.py`) which:

1. asks the user to **input a filename** (supposedly in the same folder as your program)
2. **reads the numbers** (type `int`) in the file (filename given by the user in previous step) and **stores them into a list**
3. **prints** the **minimum**, the **maximum** and the **average** of the numbers (2 decimal places for the average)

Note: your program should be **impossible to crash**. **Think about all the possibilities to make your program crash**. If one of the steps throws an exception, display a message to the user and your program should loop back to step 1.

Sample example (the user input is in red, the printed output is in blue):

```
Filename:
Number.txt
File not found
Filename:
Numbers.txt
Minimum: 12
Maximum: 21
Average: 16.50
```