Assignment 2

COMP 208 Fall 2021

due: Oct 18, 23:59

Important Instructions

- For each question in this assignment, write code in a Python file with correct filename as per the question.
- Submit your files on codePost (https://codepost.io/), and not on myCourses, email or Ed Discussion.
- Write your name and student ID at the top of each program.
- **You must do this assignment individually.** Do not use any pieces of code you did not write yourself.
- **Multiple submissions** are allowed on codePost until due date (plus 48 hours, see late submission policy below).
- **Late submission** will be accepted up to 2 days (48 hours) after the due date and will be penalized by 10% per day. Note that submitting 30 minutes late is the same as submitting 23 hours late. Late penalty will apply irrespective of the reason, be it wrong file submitted, laptop issues or any such reasons.
- **Grading.** For each question, your program will be automatically tested on a variety of test cases which will be run after due date. The test cases you will see on codePost before due date do not have any points; they are only there to help you avoid simple errors.
  - For each question, **up to 20% of the mark may be deducted** by the TA for poor code style: (i) not naming variables descriptively when appropriate; (ii) no comments in the program; (iii) very long lines or too many blanks lines; (iv) unnecessarily complicated or duplicated code. Refer to the style guide on myCourses for further details.
  - Make sure that your code runs. i.e. there are no Syntax Errors. In general, code with errors will receive a very low mark.

Question 1 — Weather Statistics [25 points]

Filename: `weather.py`

The Montreal weather station measures the weather several times over a week. For this question, you need to write a program that accepts multiple lines of space-separated numbers, where each number is a sensor reading for temperature. Your program should ask for the readings repeatedly until the user enters "stop" on a single line. Then, your program should print summary that includes average temperature, top 3 maximum temperatures and top 3 minimum temperatures. You must round all the values to 1 digit after decimal point before printing.

All readings less than -50, or greater than 50 degrees should be considered as errors in sensor and should not be used in the calculations.

Make sure that your program output matches the examples below exactly i.e. there are not typos, otherwise tests will fail on codePost.

Examples (as executed in Thonny)

Example 1:

```
>>> %Run weather.py
Enter readings: 17 28.5 0 2 4 -12.8
Enter readings: 38 39 9 29 -30 -20 9 0 0 3
Enter readings: stop
Average temperature: 7.2
Min 3 temperatures:
-30.0
-20.0
-12.8
Max 3 temperatures:
39.0
38.0
29.0
```

EXAMPLE 2: Here, incorrect readings 75 and −60 are skipped and not used in the calculations.

```
>>> %Run weather.py
Enter readings: 10.5 23 75
Enter readings: -60 20.8
Enter readings: stop
Average temperature: 18.1
Min 3 temperatures:
10.5
20.8
23.0
Max 3 temperatures:
23.0
20.8
10.5
```

Question 2 — Triples [25 points]

Filename: `triples.py`

Write a program that takes as input an integer n, and print all triples a, b, c such that:

- $1 < a < b < c < n$
- $a^2 + b^2 = c^2$
- $gcd(a, b, c) = 1$

You can assume that the user will enter a value of n such that $10 \leq n \leq 500$, and that input will always be digits (an integer) and not other letters.

Each triple a, b, c should be printed on a single line.

The greatest common divisor (GCD) of three numbers can be computed using `gcd` function in math module.

Examples (as executed in Thonny)

Example 1:

```
>>> %Run triples.py
Enter n: 10
3 4 5
```

Example 2:

```
>>> %Run triples.py
Enter n: 20
3 4 5
5 12 13
8 15 17
```

Example 3:

```
>>> %Run triples.py
Enter n: 30
3 4 5
5 12 13
7 24 25
8 15 17
20 21 29
```