UNIVERSITY OF WATERLOO

Instructions

Due date: Thursday September 15th, 2022 at 10:00am ET

This assignment covers everything up to and including Module 1.

Submission

All solutions must be submitted to MarkUs. No solutions will be accepted through email, even if you are having issues with MarkUs. If you are having issues with MarkUs, contact your instructor and/or MarkUs. Using MarkUs and your basic test results, verify that your files were properly submitted and are readable on MarkUs. Please note that MarkUs is integrated directly into EdX (see below).

Style Guide

For full style marks, your program must follow the Style Guide.

- Natural numbers in this course begin at 0. Positive integers start at 1.
- Required functions need all design recipe elements. Functions you define (e.g., helper functions) need all design recipe elements except for examples and tests.

Testing

- Testing must be done using the `check` module.
- When a function produces a floating point value, you *must* use `check.within` for your testing. Unless told otherwise, use a tolerance of 0.00001 in your tests.
- Test data for all questions will always meet the stated assumptions for consumed values.

Restrictions

Do not import any modules other than `math` and `check`. You are always allowed to define your own helper/wrapper functions, as long as they meet the assignment restrictions. Do **not** use Python constructs from later modules (e.g., `if` statements, dictionaries, loops (`for`, `while`, or others), `zip`, functions with default parameters, `sorted`, anything with `sets` or `enumerators`, slicing, indexing (square brackets), string methods, and/or lists, `ord`, `chr`, `try` and `except`). Use only the functions, methods, operations, constants and keywords as follows:

- `abs`, `len`, `max`, and `min` (however keyword parameters for these functions are not allowed)
- Any method or constant in the `math` module
- Any basic arithmetic operation (`+, -, *, /, //, %, **`)
- These typecasting operators: `int(), str(), float()`, and `type()`

Additional Notes

- While you may use global *constants* in your solutions, **do not** use global *variables* for anything other than testing.
- Read each question carefully for additional restrictions.
- **The solutions you submit must be entirely your own work. Do not look up either full or partial solutions on the Internet or in printed sources.**

Assignment Editor

Our assignment editor has some different features than other interactive blocks we have seen in this course. There are a few buttons we should describe:

- **Run Code** This button will run your code locally - you should use this when you have written some code and tests. This button also saves your code so you can revert to a previous version if needed.
- **Reset Code** This button will reset your code.
- **Show History** This button will show previous version of your code you can use to revert back to. Code is also saved every 5 minutes (but safest bet is to run your code frequently!)
- **Visualize** This will open the visualizer for your code.
- **Submit Code** This will submit your code for testing on MarkUs.
- **View Submissions** This will show you your progress on submissions to MarkUs. Note that it is your responsibility to make sure your code passes the basic tests and extensions will not be given for code that fails the basic tests.