

Assignment 3: Input, output

General instructions

- Be sure to use a plain text editor (i.e., NOT Word) when writing your code. If using `replit`, that is already the case; on Windows, Notepad++ is a good choice, while Xcode is good on Macs.
- Follow instructions in each problem on what files need to be submitted and how they should be named so the Autograder can recognize them.
- For parts that require written explanation, use the `print()` function to print your answers to the screen when the script is run.
- Make sure that your scripts run without error in order to get credit. Do not hesitate to ask for help if needed!
- Check the output of the Autograder for any issue that should be fixed. In case of “Unexpected error”, email me so I can take a look as it may be an issue with the Autograder rather than your code.
- Take ownership of your learning! Remember that you are responsible for the work you turn in. Simply copying somebody else’s answers, copying from the Internet, using AI to generate your code, sharing your code (or part of your code) in any way, or copying it from someone else will be considered academic dishonesty. Please, contact me if you have any questions about collaborations.

Problem 1

Write a code that will ask the user for her/his name, prints `Hello name!` (where `name` is the user’s name), then asks the user to enter two numbers `x` and `y` (one at a time).

The code should then print both numbers, their sum, difference, product and ratio, one per line, with text showing what is printed (for example, `x*y = 43.234`). Use f-strings to print the numbers so that they have 2 digits after the decimal point and are aligned at the decimal point (at least as long as the numbers are not too large or too small, say less than 1000 but more than 0.1).

What to submit: For this problem, you just need to submit the Python script, using the now usual name convention `youname_hw03-p1.py`.

Problem 2

Write a code that will take 3 numbers from the command-line, save them in variables `y0`, `v0`, `th0` and then use them to plot the corresponding projectile trajectory.

Make sure you don’t ask the user for the number, but use command-line arguments (i.e. argument given as in `python3 code.py 4.0 25 30`, or through the `CMDLINE` variable in the makefile).

Use f-strings to print those values on the plot, either as a title or a legend, each with 1 digit after the decimal point. For example, you should have something like `y0 = 4.0 m`, `v0 = 25.0 m/s`, `th0 = 30.0 deg`.

As an additional (optional) challenge, see if you can include the 0 in the variables as indices (as in v_0) and replace `deg` with the degree symbol (30.0°).

What to submit: For this problem, you need to submit the Python script and the figure created by your script, using the now usual name convention `youname_hw03-p2.py` and `youname_hw03-p2-fig.py`.

Problem 3

Download the file `hw03-data.txt` from Canvas. It contains 4 columns of numbers with some text on the first lines.

- (a) Load the first, second and fourth columns of this file into, respectively, variables `x`, `y`, and `z`.

- (b) Then create a plot that shows y , z and z/y as functions of x . Make sure to add labels to make clear which curve is which and to include axis labels and a plot title.
- (c) Finally, calculate the sum and average of each array x , y , and z and write the results into a file `output-p3.txt`. The content of the file should look something like

```
sum of array x is 23453.32, average = 4.345
sum of array y ...
```

Use f-strings to format the numbers so they look like the example.

What to submit: For this problem, you need to submit

- the Python script `youname_hw03-p3.py`,
- the figure `youname_hw03-p3-fig.pdf` created by your script, and
- the output file `output-p3.txt` created by your file.