

MPCS 51042-2, Python Programming

Week 1 Assignment

Due: October 1 at 11:59pm CT

For each problem, you are to submit a file named `problem<N>.py` where `<N>` is the number of the problem (e.g. `problem1.py`).

1 Installing Python and git

Install Python 3 on your computer. It is highly recommended to install Python 3.6 if possible and to use a complete distribution like [Anaconda](#) that includes third-party packages. Write a program that prints “Hello, world!” and confirm that you can run the program from a command line. It is not necessary to submit your solution to this problem (if you were able to complete the other problems, we will assume you successfully installed Python!).

To submit your assignment, you will need to have git installed on your computer. On Linux, git can easily be installed via your distribution’s package manager (`apt`, `yum`, etc.). On macOS, git is installed as part of the Xcode Command Line Tools (if you want a more up-to-date version, binaries are available from [git-scm.com](#)). On Windows, we recommend installing the full version of [cmder](#) which includes git-for-windows. On macOS and Windows, [GitHub Desktop](#) makes working with repositories on GitHub especially easy.

2 User input

Write a program that asks the user for two lists of numbers separated by commas and prints out a list of numbers that appear in both lists (no duplicates). Make sure the program works when the lists have different lengths.

Sample input/output:

```
Enter list 1: 3, 10, 12, -4, 20, 1000
Enter list 2: 5, 6, 3, 20, 4, 0, 10
[3, 20, 10]
```

The `input()` function in Python 3 allows you to accept input from the user on the command line.

3 Simple statistics

Given a file with a single column of numbers, write a program that calculates the mean, sample standard deviation, and the mode of the numbers. The name of the file should be given as the first command-line argument which can be accessed using `sys.argv[1]`. You are not allowed to use any standard-library or third-party packages that perform these calculations automatically. However, you may use data structures defined in the `collections` package from the standard library if you so wish.

Example file:

```
9
5
14
5
20
7
```

Expected output:

```
Mean: 10.0
Standard deviation: 5.932958789676531
Mode: 5
```

Given a set of numbers x_1, x_2, \dots, x_N , the mean is defined as

$$\bar{x} = \sum_{i=1}^N x_i$$

and the sample standard deviation is defined as

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}.$$

The mode is the value that occurs most often.

4 Number puzzle

Write a program that determines all six-digit numbers SLAYER (where each letter stands for the digit in the position shown) where the following equation true: $\text{SLAYER} + \text{SLAYER} + \text{SLAYER} = \text{LAYERS}$. For example, one number which satisfies this is 142857. The program should print out the above equation for each solution.

Sample output:

```
142857 + 142857 + 142857 = 428571
...
```

HINT: Although there may be some way to methodically solve the puzzle, it is by far easier to just brute force it by checking every single possibility. The `itertools.permutations` function can give you each permutation of the digits 0-9.

5 Data analysis

The City of Chicago publishes lots of [open data](#) for citizens like us to analyze. For this problem, you are provided a comma-separated value file `employees.csv` which contains information on all public employees in Chicago, their occupations, their departments, and their annual salary or hourly pay, whichever is applicable. Write a program that prints out the names and salaries of N highest paid (salaried) employees of Chicago. N should be given as a command line argument.

Although we have not covered it yet, you are allowed to use the `csv` standard library module if you wish. Otherwise, using normal file/strings methods will suffice. The csv module helps us deal with the fact that the first column has values containing commas, but is not strictly necessary here.