

Introduction to Python

Day Three Exercises

Stephanie Spielman

Email: stephanie.spielman@gmail.com

1 For-loop and if/else

1. A professor has decided to curve grades in a very special way: grades above 95 are reduced by 10%, grades between 75-95 (inclusive) remain the same, and grades below 75 are raised by 10%. You have been tasked with crunching the numbers.

- (a) Create a list of new grades that reflects these rules from the following grade list:

```
grades = [45, 94, 25, 68, 88, 95, 72, 79, 91, 82, 53, 66, 58]
```

- (b) The professor has changed his mind: he now wants to use a scaling factor of 0.078325 (instead of 0.1), because why not! Recompute the grades from part 1 using this new scaling.

- (c) The *nested* list below contains three sets of grades for silly professor's three classes:

```
all_grades = [[45, 94, 25, 68, 88, 95, 72, 79, 91, 82, 53, 66, 58], [23, 46, 17, 67, 55, 42, 31, 73], [91, 83, 79, 76, 82, 91, 95, 77, 82, 77]]
```

Create a new nested list with the curved grades for each of these groups.

- (d) Now, imagine that those three sets of grades correspond, in order, to the classes indicated in this list:

```
class_names = ["Psychology 101", "Sociology 101", "Political Science 101"]
```

Create a *dictionary* representing the *curved* grades for each of these classes. Your final dictionary should have the class name as keys, and each list of curved grades as values. **Do not recompute the curved grades** for this question. Instead, use only the nested list of curves grades and this class names list to create the final dictionary.

2. This dictionary provides the molecular weight for all amino acids:

```
amino_weights = {"A":89.09, "R":174.20, "N":132.12, "D":133.10, "C":121.15, "Q":146.15, "E":147.13, "G":75.07, "H":155.16, "I":131.17, "L":131.17, "K":146.19, "M":149.21, "F":165.19, "P":115.13, "S":105.09, "T":119.12, "W":204.23, "Y":181.19, "V":117.15}
```

Perform the following tasks with this dictionary (for ease, copy/paste it into a python script):

- Determine the molecular weight for this protein sequence:
"GAHYADPLVKMPWRTHC".
- This protein sequence, "KLSJXXFOWXNNCP" contains some ambiguous amino acids, coded by "X" and "J". Calculate the molecular weight for this protein sequence. To compute a weight for an ambiguous amino acid "X" and "J", use the *average* amino acid weight.
Hint: the `len()` and `sum()` functions and the `.values()` dictionary method will be useful! The `len()` and `sum()` functions can be used together to compute a mean value of a list. In addition, the python code `in`, which we have used for looping, can also be used to check if a certain key is in a dictionary. For example, `"X" in amino_weights` would return `False`.