1.

Write a function called phone\_numbers that takes two parameters: a file with names in subsequent lines, and a file with numbers in subsequent lines. The two files must have the same number of lines and the names and numbers must correspond one to one (if not, your function should raise an exception and ask for valid input files). The function should use these two files to create a dictionary that maps each name to its phone number. The first name in names file becomes a key in the dictionary, and the first number in numbers file becomes the value corresponding to the first name, and so on. Then, the function will return the resulting dictionary.

2.

Write a function called clean\_dict. clean\_dict() takes one parameter, a dictionary. The dictionary's keys are people's last names, and the dictionary's values are people's first names. For example, the key "Simon" would have the value "Daniel". clean\_dict() should delete any key-value pair for which the key's first letter is not capitalized. For example, the key-value pair "simon": "Daniel" would be deleted, but the key-value pair "Simon": "daniel" would not be deleted. Then, return the modified dictionary.

3.

Write a function that simulates the process of identifying the maximum value in a collection of integers. Your program must choose 100 random integers between 1 and 100, and find the maximum value in the set. Since you don’t have a lot of RAM at your disposal, you are only allowed to use 2 bytes for this program (meaning that at any given point, your program cannot store more than two integers between 1 and 100). Good luck!

4.

Write a function reverse\_list() that takes a list as parameter and then reverses the list in place and returns the reversed list. Instead of creating a new list, your program must modify the original list to reverse it. (NOTE: Please do NOT directly call the reverse() function)

5.

Write a function i\_hate\_punctuation that takes a string of text as its only parameter. Your function should return a list of the words in the string with the punctuation marks at the edges of the words removed. The punctuation marks that you must remove include commas, periods, question marks, hyphens, apostrophes, exclamation points, colons, and semicolons. Do not remove punctuation marks that appear in the middle of a word, such as the apostrophes used to form a contraction. For example, if your function is provided with the string “*Examples of contractions include: don’t, isn’t, and wouldn’t.*” then your function should return the list *[“Examples”, “of”, “contractions”, “include”, “don’t”, “isn’t”, “and”, “wouldn’t”]*