

Week 6 Assignment

While many of the exercises in this chapter can be solved with lists or if statements, most (or even all) of them have solutions that are well suited to dictionaries. As a result, you should use dictionaries to solve all of these exercises instead of (or in addition to) using the Python features that you have been introduced to previously.

Question 1

Write a function named `reverseLookup` that finds all of the keys in a dictionary that map to a specific value. The function will take the dictionary and the value to search for as its only parameters. It will return a (possibly empty) list of keys from the dictionary that map to the provided value.

Include a main program that demonstrates the `reverseLookup` function as part of your solution to this exercise. Your program should create a dictionary and then show that the `reverseLookup` function works correctly when it returns multiple keys, a single key, and no keys. Ensure that your main program only runs when the file containing your solution to this exercise has not been imported into another program.

Question 2

In this exercise you will simulate 1,000 rolls of two dice. Begin by writing a function that simulates rolling a pair of six-sided dice. Your function will not take any parameters. It will return the total that was rolled on two dice as its only result. Write a main program that uses your function to simulate rolling two six-sided dice 1,000 times. As your program runs, it should count the number of times that each total occurs. Then it should display a table that summarizes this data. Express the frequency for each total as a percentage of the number of rolls performed. Your program should also display the percentage expected by probability theory for each total. Sample output is shown below.

Total	Simulated Percent	Expected Percent
2	2.90	2.78
3	6.90	5.56
4	9.40	8.33
5	11.90	11.11
6	14.20	13.89
7	14.20	16.67
8	15.00	13.89
9	10.50	11.11
10	7.90	8.33
11	4.50	5.56
12	2.60	2.78

Question 3

Morse code is an encoding scheme that uses dashes and dots to represent digits and letters. In this exercise, you will write a program that uses a dictionary to store the mapping from these symbols to Morse code. Use a period to represent a dot, and a hyphen to represent a dash. The mapping from characters to dashes and dots is shown in the table below.

Your program should read a message from the user. Then it should translate all of the letters and digits in the message to Morse code, leaving a space between each sequence of dashes and dots. Your program should ignore any characters that are not listed in the previous table. The Morse code for Hello, World! is shown below:

• • • • • • — • • • — • • — — — • — — — — — • — • • — • • — • • — • •

Character	Code	Character	Code	Character	Code	Character	Code
A	. -	J	. - - -	S	. . .	1	. - - - -
B	- . . .	K	- . -	T	-	2	. . - - -
C	- . .	L	. - . .	U	. . -	3	. . . - -
D	- . .	M	- -	V	. . . -	4 -
E	.	N	- .	W	. - -	5
F	O	- - -	X	- . . -	6	-
G	- - .	P	. - - .	Y	- . - -	7	- - . . .
H	Q	- - . -	Z	- - . .	8	- - - . .
I	. .	R	. - .	0	- - - - -	9	- - - - .

Question 4

Create a program that determines and displays the number of unique characters in a string entered by the user. For example, Hello, World! has 10 unique characters while zzz has only one unique character. Use a dictionary or set to solve this problem.

Question 5

Two words are anagrams if they contain all of the same letters, but in a different order. For example, "evil" and "live" are anagrams because each contains one "e", one "i", one "l", and one "v". Create a program that reads two strings from the user, determines whether or not they are anagrams, and reports the result.