A) Creating, filling and accessing dictionaries

Exercise A1)

Write a program that contains a hard-coded dictionary of phone contacts (key: contact name, value: contact number) populated with dummy data. Your program should:

- Ask the user to enter the name of the contact they wish to view.
- Retrieve the corresponding phone number for that contact and display it to the user.

Your program should not handle where the contact name is not present in the dictionary.

Exercise A2)

Write a program that contains two dictionaries of phone contacts. In each dictionary, the key is the contact name and the value is the contact's phone number.

The program should combine these two dictionaries into a single dictionary (either a new dictionary separate from the original two, or by adding the contents of one dictionary to the other) and display it to the user. You should not consider the possibility of conflicts (where a key appears in both dictionaries) in this program.

B) Checking for keys

Exercise B1)

Amend your solution to A1 to handle the possibility of the contact name not being present. If the contact isn't present, an appropriate message should be displayed to the user.

Exercise B2)

Amend your solution to A2 to handle the possibility of a conflict. Where a key appears in both dictionaries, you should display the values from both dictionaries to the user and ask which should be retained.

C) Deleting from dictionaries

Exercise C1)

Create a csv file of user data following the format:

username,password,first name,last name

You should create this file manually, i.e. write it out in a text editor/spreadsheet. You are not required to generate it with a python program. Your file should contain at least 5 lines of user data.

Exercise C2)

Write a program that:

1. Creates a blank users dictionary.
2. Reads in your user data file and stores each line as a **list** of user information. Each list of user data should be stored in the users dictionary.
    a. When storing each line of data, the key should be the username for that line, and the value should be the list of user information.
    b. When adding to the dictionary, you should confirm the username doesn't already exist before you add a new user's information.

Note: If you don't want to do the information parsing step yourself (i.e. splitting the data up based on the commas), consider using the csv module to read it in. If you haven't tried using the csv module before, https://www.programiz.com/python-programming/csv is a nice introduction.

Exercise C3)

Amend your answer to part C2 to prevent duplicate usernames being added where the case is different, e.g. "Michelle" and "michelle" would not be allowed.

Exercise C4)

Amend your answer to part C3 to delete all user information from your dictionary where the username is shorter than 4 characters long. Note: This should be done AFTER the dictionary has been loaded, i.e. you are not preventing them from being added from the file, you need to remove them after the file has been fully processed and its contents stored in the dictionary.

D) General problem-solving using dictionaries

Exercise D1)

Write a program that uses a dictionary to count the number of times each word in a file appears within that file. The count should be case-sensitive.
Hint: You use a key to search for something specific in a dictionary. If the key in the dictionary is a word in the file, what could the value be?

Exercise D2)

Amend your solution to D1 to make the count case-insensitive.