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)

Amend your solution to A1 to handle where the contact name is not present. Display the contact information to the user where a matching contact can be found. Display an error message to the user if the contact name cannot be found.

Exercise A3)

Add a new section to your program that does the following:

1. Takes in a contact name from the user
2. Confirms this name is not already present in the dictionary
    a. If it is already present, informs the user and asks if they wish to enter another contact
    b. If it is not already present, takes in a contact number and adds that contact to the dictionary

This should repeat until either the user successfully adds a contact OR they choose not to try and add another one.

B) Deleting from dictionaries

Exercise B1)

Add a new section to your program that does the following:

1. Takes in a contact name from the user
2. If the name is present in the dictionary, asks the user if they wish to delete just the number, or to remove all trace of the contact
    a. If they want to just delete the number, set that contact's value to be None
    b. If they want to remove all trace, delete the key-value pair from the dictionary entirely

C) Using dictionaries for user access control

Exercise C1)

Add a new section at the start of your program that:

1. Creates a new dictionary for user credentials. The dictionary should store username - password key-value pairs.
2. Asks the user to log in by entering their username and password
3. Checks if the username exists
    a. If it does, checks the password to confirm the user can be logged in
4. If the username does not exist or the username exists but the password doesn't match the stored one, inform the user that the attempt was unsuccessful and terminate the program
5. If the attempt is successful, allow the rest of the program to continue

D) Menu-driven behaviour

Exercise D1)

Add a new section to your program that wraps the above features in a menu. The menu should repeat until the user selects an option to exit the program.

E) General problem-solving using dictionaries

Exercise E1)

Write a program that uses a dictionary to count the number of times each word in a file appears within that file. Do this first with the count case-sensitive, then amend it to be case-insensitive.

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?