Assignment 2 – Word Jumbled and Encryption, Due 9/17/25, 11:59pm

Goals

- Lists
- String processing and manipulation
- Loops
- You are to submit a single Python script that executes both parts 1 and 2

Part 1 – Word Jumble Game

- **Requirements**
  - Start: create a list with words (at least 10 words) that you will use in the game. Use **`random.choice (…)`** to pick one of the words.
  - Create a jumbled version of the word and display it to the user.
    - This part is challenging, but there is a hint below to help. You should be able to complete it with what we have covered and may not use automated methods you may find online (this means you may not use **`random.shuffle(), random.sample(),`** or similar methods)
    - Hint: For all word processing, use lists. Since strings are immutable, you cannot jumble the letters directly. However, you can make a jumbled copy of the original word. To jumble a word, you could rearrange the letters in the word. If you do this correctly, multiple tries on the same original word will end up with different jumbled versions. You **are not to** manually code a jumbled version of your word. e.g.

```
list = ["ape"]
jumbledList = ["eap"]
```

  - Have the user guess the word until they get it correct.
  - Count the number of guesses it takes.

Sample Output for Part 1

```
The jumbled word is "yhtnpo"
Please enter your guess: htnpoy
Try again.

Please enter your guess: poythn
Try again.

Please enter your guess: python
You got it!
It took you 3 tries.
```

Part 2 – Encrypt / Decrypt

- **Background**
  - The Caesar cipher (named for Julius Caesar) is a simple method for encrypting a message. This involves "shifting" each letter in the message by a fixed number.
  - For example, consider a shift of 3 letters

    Original alphabet: **abcdefghijklmnopqrstuvwxyz**

    Cipher alphabet: **defghijklmnopqrstuvwxyzabc**
  - To encrypt a message, substitute plain-text letters with corresponding cipher letter. Do **not** shift spaces or punctuation.
  - Using that new encryption key (alphabet),

    Original message: **hello world.**

    Encrypted message: **khoor zruog.**
  - To decrypt, repeat the process
- **Requirements**
  - Write a program that prompts the user for the shift value (e.g. 3, or 4, or anything less than 26) and then a plain-text message to encode. Encrypt the message and print out the encrypted message. Then use your program to decrypt the message back to its original state. Compare the decrypted message with the original.
- Hints
  - To create a cipher (shifted alphabet), first start with a list of alphabet letters as strings, and then use slicing to create the cipher.
  - Do not shift punctuation or spaces—simply leave them as is (see "hello world" from above – notice the period)
  - When you are replacing letters, notice that the letter in the original alphabet is at the same index as the cipher letter you are going to replace it with.
  - Design your algorithm on paper before coding

Sample Output for Part 2

```
Enter a message: lamb biryani
Enter a number to shift by (0-25): 3
Encrypting message....
      Encrypted message: odpe elubdql
Decrypting message....
      Decrypted message: lamb biryani
```

```
Original message: lamb biryani
```

Deliverables and Submission Instructions

- Create a zip file of your Python source code and upload the zip file to Brightspace

Grading

Item	Points
Part 1: Word Jumble	15
Part 2: Encrypt / Decrypt	15
Total*	**30**

** Points will be deducted for poor code style, or improper submission.*