CSCI 220L

Lab 8

Learning Objectives

- Practice string formatting in Python.
- Use `ord()` and `chr()` to encode and decode strings.
- Read and write data in files.

Activities

Complete the attached Python file `lab8.py` with the following functions. Make sure to use meaningful variable names, input prompts, and output messages.

1. String Formatting

   The function `format_practice()` contains several print statements. Fill in the blanks with code that will make statement print the expected output.

   (a)
   ```
   print("The value of {} is {}.".format(_____))
      # Expected: The value of x is 19.
   ```

   (b)
   ```
   print("It's raining {1} and {0}".format(_____))
      # Expected: It's raining cats and dogs
   ```

   (c)
   ```
   print("My name is {0}, {1} {0}".format(_____))
      # Expected: My name is Bond, James Bond
   ```

   (d)
   ```
   print(_____.format(2.3, 0.4567))
      # Expected: 2.30 0.457
   ```

   (e)
   ```
   print(_____.format(3, 7.4589))
      # Expected: Time left: 03:07.46
   ```

   (f)
   ```
   print(_____.format("Steph", "Curry", 43.75432))
      # Expected: Steph Curry: 43.75
   ```

   There is another way to do string formatting in Python called f-strings, which were added to Python in 2016 (version 3.6). Some programmers (myself included!) prefer f-strings to the `format()` method because they are more concise. Here’s an article about how to use them if this interests you.

2. Naming Conventions

   Write a function `rename()` that asks the user to enter a string in lower_snake_case and displays it back in lowerCamelCase.

3. Encryption Technique I: Caesar Cipher

   A Caesar cipher is a simple substitution cipher where each letter of a message is shifted by a fixed amount, called the key. For example, the message “HELLO” encrypted with a key of 2 would produce the ciphertext “JGNNQ”. The original message can be recovered by shifting the ciphertext by the negative of the key.

   (a) Write a function `caesar()` that asks the user to enter a message and an integer key, and then encrypts the message with the key by shifting it through the ASCII character set. You can use `ord()` to convert a single character into a number so that it can be shifted, and then use `chr()` to convert the resulting number back into a character. Don’t worry about characters that get shifted past Z.

   (b) The attached file `secret.txt` contains a ciphertext that has been encrypted with a special Caesar cipher where the key is the length of the message. Write a function `secret()` that reads the contents of the file, decrypts it, and displays the recovered message.

   (c) You may have noticed that using a large key produces a ciphertext with non-letter characters. A wrapping Caesar cipher is one in which characters shifted past Z wrap back around to A. For example, the message “XYZ” encrypted with a key of 3 would produce the ciphertext “ABC”. Write a function `wrapping()` that works as in part (a) except that it performs a wrapping Caesar cipher. You may assume the message contains only uppercase letters.

   Tip: You can import `string` and use `string.ascii_uppercase` as your codebook, but you can also get by with just `ord()` and `chr()`.

4. Encryption Technique II: Book Cipher

   A book cipher uses the contents of a book owned by both the sender and receiver as the key for encrypting a message. As the Wikipedia article mentions, two popular types of books used as keys are Bibles and dictionaries.

   (a) You have been tasked with decrypting the following ciphertext, which appears to be a binary string partitioned into 6-bit segments:

```
011110 011111 000011 001101 000000 000011 000101 011101 001110
```

   You have also recovered a Bible with certain words highlighted. When compiled together and sorted, the following list was produced:

```
['ANYTHING', 'AS', 'BUT', 'CAN', 'COUNT', 'CREATE', 'DIFFICULTY', 'DO', 'DOES',
'END', 'EVERY', 'GO', 'HOW', 'IMAGINE', 'IMPOSSIBLE', 'IN', 'IS', 'IT', 'LIES',
'LIFE', 'LONG', 'MATTER', 'MIDDLE', 'NOT', 'OF', 'OPPORTUNITY', 'SLOWLY', 'STOP',
'THAT', 'THE', 'THOSE', 'WHO', 'YEARS', 'YOU', 'YOUR']
```

   You believe this is a form of substitution cipher. You can decipher the code by interpreting each segment as an integer and matching it to its position in the list. For example, 000000 refers to the first word in the list, 000001 refers to the second, and so on. Write a function `book_cipher()` that decrypts the ciphertext and displays the original message.

Hint: The provided function `binary_list()` creates a list of unsigned binary numbers for you so that you can “convert” a number from binary to decimal by finding it in the list. For example, `code = binary_list(10)` produces a list of the first 10 binary numbers (0000 to 1001). What size of list do you need to store every possible 6 bit number?

(b) You are a military contractor that has been tasked with decrypting the following ciphertext, which appears to be a list of integers separated by dashes:

```
22933-22372-77345-171880
```

You have reason to believe that the codebook is an English dictionary. Write a function `dict_cipher()` that reads the attached `dictionary.txt` into a list, uses it to decrypt the ciphertext, and write the result to a new file called `message.txt`.

Upload `lab8.py` to the OAKS dropbox before the deadline. Make sure you have most of the exercises completed before your lab meeting.