COSC 145 Intro to Python Prog.
Fall 2023
Assignment 6

Due Date: November 02, 2023

Students are strictly prohibited from using ChatGPT or any other artificial intelligence tools to complete this assignment. All submissions should be original work.

Please submit five Python files (.py), one for each question, plus an optional file named extra-credit.txt if you completed the extra credit section of question 5.

If you have any questions, please don't hesitate to ask in class, during office hours, via email, or by setting up an appointment. I am always here to help.

Question 1 (10 points):

A prime number is a number that is only evenly divisible by itself and 1. For example, the number 5 is prime because it can only be evenly divided by 1 and 5. The number 6, however, is not prime because it can be divided evenly by 1, 2, 3, and 6. Write a Boolean function named is_prime which takes an integer as an argument and returns true if the argument is a prime number, or false otherwise. Use the function in a program that prompts the user to enter a number and then displays a message indicating whether the number is prime.

Question 2 (10 points):

Write a program that reads a string from the user containing a date in the form mm/dd/yyyy. It should print the date in the format March 12, 2018.

Question 3 (15 points):

Write a program with a function that accepts a string as an argument and returns a copy of the string with the first character of each sentence capitalized. For instance, if the argument is hello. my name is Joe. what is your name? the function should return the string Hello. My name is Joe. What is your name? The program should let the user enter a string and then pass it to the function. The modified string should be displayed.

Question 4 (20 points):

Write a program that lets the user play the game of Rock, Paper, Scissors against the computer. The program should work as follows:

1. When the program begins, a random number in the range of 1 through 3 is generated. Depending on the number:
    - If the number is 1, the computer chooses *rock*.
    - If the number is 2, the computer chooses *paper*.
    - If the number is 3, the computer chooses *scissors*.

    Note: Dont display the computers choice yet.
2. The user enters his or her choice of rock, paper, or scissors at the keyboard.
3. The computers choice is then displayed.
4. A winner is selected based on the following rules:
    - If one player chooses rock and the other scissors, then *rock* wins. (The rock smashes the scissors.)
    - If one player chooses scissors and the other paper, then *scissors* wins. (Scissors cuts paper.)
    - If one player chooses paper and the other rock, then *paper* wins. (Paper wraps rock.)
    - If both players make the same choice, the game must be played again to determine the winner.

Note: You should implement at least three functions as deemed necessary for the problem. For instance, one function could generate a random number between 1 and 3 and return its corresponding choice: rock, paper, or scissors. Feel free to design any other functions that might be useful for this task.

Question 5 (45 points)

Objective

This question is designed for you to practice working with lists, files, and functions.

Scenario

Consider two individuals, Alice and Bob. They aim to exchange messages in a secure manner, ensuring that no third party can intercept or decipher their conversation. To accomplish this, they opt for a Book-Based Cipher method.

A Book-Based Cipher requires both the sender and receiver to have an identical copy of a book or text, serving as their shared key. Let's understand how Alice and Bob employ this method:

Encoding Process

1. Alice selects a message to convey to Bob.
2. For every word in her message, she searches for it within their shared book.
3. She then records the page, line, and word position of each word. For example, if "moon" is found on page 45, line 7, as the second word, it would be encoded as "45.7.2".
4. Alice subsequently sends this numeric sequence to Bob as the encoded form of her message.

Decoding Process

1. On receiving the encoded message, Bob refers to his copy of the book.
2. He follows each given location (page, line, word) in sequence.
3. This allows Bob to piece together and understand Alice's original message.

Assignment Requirements

1. **Reading a File:** Your program should be able to read a text file (a book) into a list of strings, a string, or any other data structure. We've covered reading files in previous classes, but here are a couple of quick examples:

   Example: Read a file into a list (each line as a separate item):

   ```python
   with open('filename.txt', 'r') as file:
       data_list = file.readlines()
   print(data_list) # verify file reading
   ```

   Example: Read a file into a string:

   ```python
   with open('filename.txt', 'r') as file:
       data_string = file.read()
   print(data_string) # verify file reading
   ```

   For testing, you should use the text file book1.txt, which is attached to this assignment.

2. **Character Removal:** Assuming you have a list containing strings, where each string represents a line, remove all punctuation, special, and newline characters from each string. Here are the characters that should be removed from each string:

   - \n (Newline)
   - : (Colon)
   - . (Period)

- , (Comma)
- ' (Apostrophe)
- - (Hyphen)
- ? (Question mark)
- ((Open parenthesis)
-) (Close parenthesis)

Bear in mind that strings will still retain spaces. Also, the list may have empty strings.

Example: Remove newline characters "\n" and commas from strings in a List:

We have a list containing two strings. The objective is to remove newline characters '\n' and commas from the strings.

```python
# Sample list containing two strings
data_list = ["Hello, World\n", "Python, Programming\n"]

# Iterate over each string in the data_list and clean it in-place
for i in range(len(data_list)):
    data_list[i] = data_list[i].replace('\n', '').replace(',', '')

# Print the cleaned list
print(data_list)
```

This will produce the output:

```
['Hello World', 'Python Programming']
```

For this assignment, you should create a function called *clean_data* that receives a list of strings, removes the characters mentioned above from it, and returns the cleaned list.

3. **Encryption function:** Your program must contain a function named *encrypt_word()*. This function takes the cleaned list of strings and a word, then returns its location in the format "page#.line#.word#". Assume each page contains 10 lines/strings. Even if the word is found in multiple strings, you can choose any occurrence to encode. For instance, if the function is given the list of strings and the word "moon" and this word appears third on line 5 and fifth on line 11 the function could return either "1.5.3" or "2.1.5". Remember, line 11 translates to line 1 on page 2.

4. **Decryption function:** Your program should include a function named *decrypt_location().* This function accepts the cleaned list of strings and a location in the format "page#.line#.word#", then retrieves the word at that specific location. For example, if the function is provided with the list of strings and the location "2.1.5", it should return the word "moon" since this word is the fifth word on line 1 of page 2.

5. **Program Execution:** When you run your program, it should offer three choices to the user:

   (a) Encrypt a message
   (b) Decrypt a message
   (c) Exit the program

   - If "Encrypt" is selected, the user will be prompted to input plain text (e.g., "hello world"). The program should then display the locations of these words from the book. If a word is found, its location will be shown, such as "1.2.7". If the word is not found in the book, the program should display "?.?.?" for that word. For instance, if "hello" is found at "1.2.7" and "world" is found at "2.5.3", the output would be "1.2.7 2.5.3". However, if "hello" is found but "world" is not, the output would be "1.2.7 ?.?.?".
   - If "Decrypt" is selected, the user will be prompted to input locations, such as "1.2.7 2.5.3". The program will then display the corresponding text. If a given location is valid and matches a word in the book, that word will be shown, e.g., "hello". However, if the location is invalid, the program will display "???" for that specific location. As an illustration, if "1.2.7" matches the word "hello" but "200.5.3" is invalid because the text only has 15 pages, the output will be "hello ???".
   - If "Exit" is chosen, the program should terminate.

6. **Extra points, testing with other students (10 points):** Encrypt a message using your program and send the encrypted message (locations) to another student by email, text, or any other method. The other student receives your encrypted message and decrypts it using their program (not yours) and replies with the plain text (original message) to you. If you have done this, create a file called extra-credit.txt and mention your name, the other student's name, your original text, its encryption, and whether or not the other student was able to decrypt it or not. **Important: both of you should have the same copy of the textbook, which is attached to this assignment (book1.txt)**.

7. **Grading Criteria:**

   - **clean_data function (5 points)**
   - **encrypt_word function (10 points)**
   - **decrypt_location function (10 points)**
   - **main function and program execution (15 points)**