Homework 4: Loops & I/O

L555

Due Wednesday, September 26

1. This program from a hypothetical student is supposed to read in 50 words, put them into a list, sort them, and then print every one with a count of how many more words are to be printed. Unfortunately, there are 5 mistakes in the program. Find them and correct them.

```python
words = {}
for x < 50:
    words.append(raw_input('next word: '))
words.sorted()
for x in range (1,50):
    print words[x], 'words to do:', 50-x
```

2. A Fibonacci number is calculated by taking the sum of the previous two numbers, starting from 0 and 1. So, the first ten Fibonacci numbers are: 0, 1, 2, 3, 5, 8, 13, 21, 34.

   (a) Write a program that prompts a user for a number \((n)\) and uses a `while` loop to calculate the \(n^{th}\) Fibonacci number. Remember that \(f_i = f_{i-2} + f_{i-1}\) and so you’ll need variables to account for this.

   (b) Write an alternative version of the program which uses a `for` loop instead.

3. Write a program that reads in 100 words from the terminal and creates a frequency lexicon. Store the words in a list and the frequency counts in another list so that the frequency count for a specific word has the same position as the word. Make sure that if a word is already in the list, it should not be added, but only the frequency count must be increased. Finally, output the list.

To test your program use the file wordlist100.txt from oncourse under Resources/data.

4. For this question, I want you to start working on the problem and turn in something showing where you’re at. It does not have to be complete or even pretty, but must show evidence of thought. You’ll finish it for next week’s assignment.

You’re going to have a user enter a sequence of \(a\)’s and \(b\)’s one at a time. Anything other than an \(a\) or a \(b\) signals that they are done inputting.

- At the very minimum, they must enter \(ab\). If they don’t, tell them “I’m sorry; that is not a valid string.”
- They can have as many \(b\)’s as they want, but every time they enter an \(a\), it must be followed by a \(b\). If they enter anything other than a \(b\) after an \(a\), stop asking for input and tell them “I’m sorry; that is not a valid string.”
- They must stop at a \(b\), i.e., the string cannot end in \(a\). If you are familiar with finite-state machines, this is the same as what is in Figure 1.
- When the user enters a character other than \(a\) or \(b\), evaluate whether the string is valid and either print “Thank you!” or “I’m sorry; that is not a valid string.”
Here are some example sessions:

Please enter a letter: a
Please enter a letter: b
Please enter a letter: b
Please enter a letter: END
Thank you!

Please enter a letter: a
Please enter a letter: a
I’m sorry; that is not a valid string.

Please enter a letter: a
Please enter a letter: b
Please enter a letter: b
Please enter a letter: a
Please enter a letter: END
I’m sorry; that is not a valid string.

Please enter a letter: a
Please enter a letter: b
Please enter a letter: a
Please enter a letter: b
Please enter a letter: a
Please enter a letter: b
Please enter a letter: c
Thank you!