Programming Exercises

The exercises in this section are optional and do not report to the performance dashboard. Instructors can decide whether to assign these exercises and students can check the correctness of their programs using the Check Exercise tool.

Sections 14.2– 14.6

***14.1** (*Display keywords*) Revise LiveExample 14.4 CountKeywords.py to display the keywords in a Python source file as well as to count the number of the keywords.

***14.2** (*Count occurrences of numbers*) Write a program that reads an unspecified number of integers and finds the ones that have the most occurrences. For example, if you enter `2 3 40 3 5 4 –3 3 3 2 0`, the number `3` occurs most often. Enter all numbers in one line. If not one but several numbers have the most occurrences, all of them should be reported. For example, since `9` and `3` appear twice in the list `9 30 3 9 3 2 4`, both occurrences should be reported.

***14.3** (*Count the occurrences of each keyword*) Write a program that reads in a Python source code file and counts the occurrence of each keyword in the file. Your program should prompt the user to enter the Python source code filename.

***14.4** (*Tkinter: Count the occurrences of each letter*) Rewrite LiveExample 14.5 using a GUI program to let the user enter the file from an entry field, as shown in Figure 14.3a. You can also select a file by clicking the Browse button to display an Open file dialog box, as shown in Figure 14.3b. The file selected is then displayed in the entry field. Clicking the Show Result button displays the result in a text widget. You need to display a message in a message box if the file does not exist.

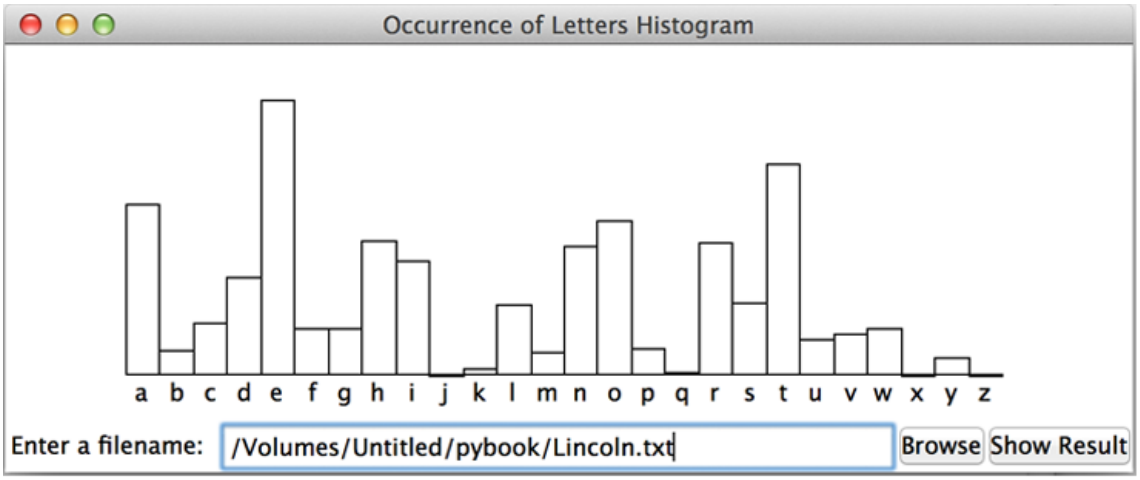
Figure 14.3

The program lets the user select a file and displays the occurrence counts of the letters in the file.

(Screenshots courtesy of Microsoft Corporation.)

***14.5** (*Tkinter: Count the occurrences of each letter*) Revise the preceding exercise to display a histogram for the result, as shown in Figure 14.4. You need to display a message in a message box if the file does not exist.

Figure 14.4

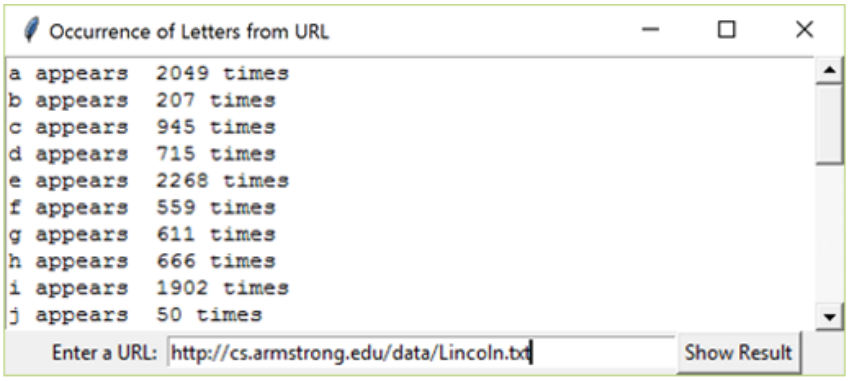


The program lets the user select a file and displays the occurrence counts in a histogram.

(Screenshot courtesy of Apple.)

***14.6** (*Tkinter: Count the occurrences of each letter*) Rewrite LiveExample 14.5 using a GUI program to let the user enter the URL from an entry field, as shown in Figure 14.5. Clicking the *Show Result* button displays the result in a text widget. You need to display a message in a message box if the URL does not exist.

Figure 14.5

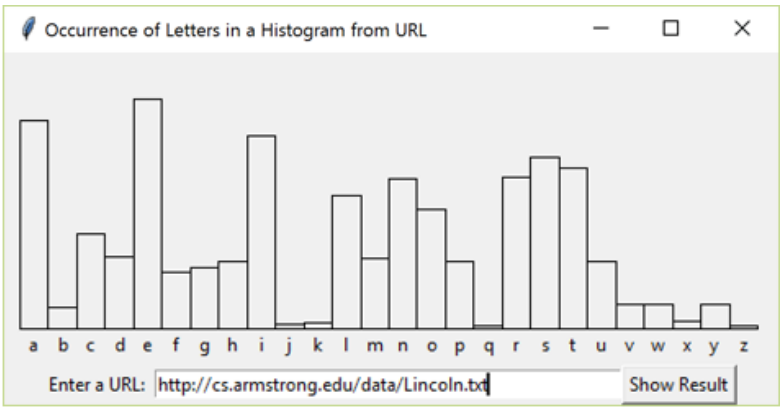


The program lets the user enter a URL for a file and displays the occurrence counts of the letters in the file.

(Screenshot courtesy of Microsoft Corporation.)

***14.7** (*Tkinter: Count the occurrences of each letter*) Revise the preceding exercise to display a histogram for the result, as shown in Figure 14.6. You need to display a message in a message box if the URL does not exist.

Figure 14.6



The program lets the user enter a URL for a file and displays the occurrence counts of the letters in a histogram.

(Screenshot courtesy of Microsoft Corporation.)

14.8 (*Display nonduplicate words in ascending order*) Write a program that prompts the user to enter a text file and reads words from the file and displays all the nonduplicate words in ascending order.

***14.9 (*Game: Hangman*) Write the hangman game with a *graphics* display, as shown in Figure 14.7. After seven misses, the program displays the word. The user can press the *Enter* key to continue to guess another word.

Figure 14.7

The hangman game lets the user enter letters to guess a word.

(Screenshots courtesy of Apple.)

*14.10 (*Guess the capitals*) Rewrite Programming Exercise 8.39 using a dictionary to store the pairs of states and capitals so that the questions are randomly displayed.

*14.11 (*Count consonants and vowels*) Write a program that prompts the user to *enter* a text filename and displays the number of vowels and consonants in the file. Use a set to store the vowels `A`, `E`, `I`, `O`, and `U`.

*14.12 (*Subtraction quiz*) Rewrite Programming Exercise 7.36 to store the answers in a set rather than a list.