

Practice Problems for Week 4

Topics: for Loops, lists, Reading from Files

APOSPWP (5th Edition) Chapter 2 Exercises 2.3, 2.6, 2.7, 2.8, 2.9, 2.11, 2.12, 2.13, 2.19, 2.22

APOSPWP (5th Edition) Chapter 4 Exercises 4.3, 4.9

1. Ask for an integer, n , from the user. Use a while loop to calculate $\sum_0^n 2n^2$
2. Ask the user to input their favourite saying. Ask the user to input a letter. Use a for loop to count the number of times that letter appears in their favourite saying, and print the count.
3. Ask the user to input their favourite saying (assume only letters and spaces are entered). Determine which letter appears most frequently in the saying, and output a message stating which letter was most frequent. Do not include spaces in the count.

Hint: Rather than looking for a's then b's then c's, etc. instead use the saying itself and only count letters that are in the saying. For example, if the user enters "all for one and one for all", count the number of a's then l's then f's then o's etc. Keep track of the character with the highest count, and update it when you find a new highest count.

Write one version where capitalization is ignored, and one where uppercase letters are counted separately from lowercase.

4. Given a height, print a triangle made of asterisks. E.g. If height is 5:

```
*  
**  
***  
****  
*****
```

If you used for loops, write a second version of the program using while loops. If you used while loops, write a second version using for loops.

5. Given a height, print an upside-down triangle made of asterisks. E.g. If height is 4:

```
****
***
**
*
```

If you used for loops, write a second version of the program using while loops. If you used while loops, write a second version using for loops.

6. Given a height, print an "M" made of asterisks. E.g. If height is 4:

```
*      *
**     **
***    ***
*****
```

7. Assume that you have a list of numbers. For example:

```
data = [8, 19, 53, 27, 31, 82, 16, 72, 3, 22]
```

Calculate the square of each number and store the squares in a second list. Calculate and then print the product of all numbers in the second list. (Recall that the product of two numbers, a and b, is multiplication, $a * b$.)

8. Assume that you have a list of numbers. For example:

```
data = [8, 19, 24, 37, 39, 41, 49, 53, 58, 25, 81, 90]
```

Traverse the list and determine if it is in ascending order.

9. Assume that you have two lists of numbers. For example:

```
listOne = [8, 19, 53, 27, 31, 82, 16, 72, 3, 22]
```

```
listTwo = [82, 74, 20, 31, 19, 22, 65, 27, 45]
```

The lists are not necessarily the same length. Count how many numbers from `listOne` are present in `listTwo`, and output the count.

10. In Spyder, create a .txt file with a few lines of text, and save the file in the same folder as your code files. Write a program that will open the text file and make a list of all unique words in the file. (You may assume that the file contains only words and spaces.) There should be no duplicate words in the list. Print the list, and the number of unique words in the file.

11. In Spyder, create a .csv (comma-separated values) file with 10-20 lines. On each line, include a course name or code, a term, and a letter grade. For example:

COMP 1012, Fall 2021, A

MATH 1510, Fall 2021, B+

PHYS 1050, Summer 2021, B

Write a program that will:

- Open the .csv file and load the data into three lists, one for course codes, one for term, and one for letter grade. Make sure you strip off extra white space as you insert the data into the lists. The data from a particular line in the file will end up at the same index in each list - a set of lists like this is referred to as *parallel* lists.
- Ask your user to enter a term.
- Look through the data for grades from the term the user specified. Calculate the grade point average for the term using the University's standard conversion of A+ = 4.5, A = 4.0, B+ = 3.5, etc. Print a message for the user such as:
Your GPA for Fall 2021 is 3.75.

Past Quiz/Test Questions:

12. Write a Write a program that will:

- Open a text file named quizData.txt. Assume that the file contains only alphabetical characters, with one or more words on each line in the file.
- Count the total number of words in the file.
- Print the number of words in the file.

13. Write a program that will:

- Ask the user to enter a number from 1 to 10, inclusive.
 - You can assume the user types a number, but should verify that the number is in the range 1 to 10. If not, ask the user to try again, until they enter a valid number.
- Ask the user to enter the number of numbers they specified. See sample output below – include which number is being requested in your prompt.
- Calculate the sum of these numbers.
 - You can assume the user types only numbers, and do not need to validate each one.
- Output the sum.

Sample output:

Please enter a number (1-10): 3

Please enter number 1: 6

Please enter number 2: 2

Please enter number 3: 7

The sum of your numbers is 15.

(green printed by your program, blue entered by the user)

14. Write a program that will count the number of words in a file:

- Assume that you have a filename ending in ".txt" stored in a variable called **userInput**.
- Open the file named in userInput and store the file pointer in a variable called **myFile**.
- Create a variable to count the number of words. Start the count at 0.
- For each line in myFile:
 - Find the number of words in that line. (Hint: Assume there are only words separated by spaces).
 - Add the number of words in the current line to the total count.
- Print out the total number of words in the file.

15. You've been given a file that has just numbers, but any number of numbers per line (always at least one - no blank lines).

Example:

111 112 13

12

12

12

42 762 556 308 3030

Write a program that opens the file, and gets a sum of all the numbers.

Print the sum (no fancy formatting needed).

16. Given a list of integer numbers

Example (but not limited to):

nums = [1, 2, 3, 4, 22, 44, 33, 2]

Calculate and print the sum of the even numbers in the list.