Homework #4

Physics 129 Spring 2022

Problems due **Saturday, April 23, at 11:55 P.M.**

Please read the homework guidelines handout on the course web page.

Before attempting this assignment, ensure your RPi is connected to the Internet, then run the `update_physrpi` script.

Better answers and code will get better grades.

Reading

→ Complete by **Monday, April 25**

- Read chapters 10–11, 19, and the material on the `printf` command in chapter 21 in Shotts.
- Read chapter 4, section 6.1, and section 10.1.1 in K&N.

Problems

1. **Factoring Numbers.** Write a program that
   a. Prompts the user to enter an integer
   b. Attempts to convert the input string to an appropriate variable
   c. Continues to prompt the user until the conversion is successful
   d. Prints out the prime factors of the number

   Please note: the point of this problem is to write the fastest possible code, not to use sophisticated math. Use only elementary division and/or remainder operations. Do not use a number field sieve, nor any other non-trivial algorithm.

   The TA will test your program on these numbers, among others:
   9879878778787
   2348923847938743.

Hints: Study the `error_handling.py` example program in `$HOME/physrpi/python/` on your RPi. Think hard about what the largest number is that you must test before you will have found all of the prime factors.

2. **Dictionaries and Databases.** Create a comma-separated values (csv) file having on each line a comma-separated list including a person's last name, first name, favorite color, favorite food, favorite field of physics, and most admired physicist. The file should contain at least three lines, each of which describes a different person. Write a program that

   a. Reads in the csv file

   b. Stores the information from the csv file in a list of Python dictionaries, one per line. You should choose the key names, for example 'last', 'first', 'color', etc.

   c. Prints a list of keys for the user

   d. Prompts the user to choose a key

   e. Prints all names in alphabetical order with the requested values. For example, if the user selects the key 'color', the program should print something like:

   ```
   Einstein, Albert:  green
   Fermi, Enrico:  red
   Michelson, Albert:  blue
   ```

   Hint: create a list of output strings and sort it before printing.

   f. Allows the user to continue selecting keys for display until he/she is finished.

3. **Execution Speed.** Write a program that uses `time.perf_counter()` to determine how long it takes each of the following to execute inside of a loop:

   a. Nothing (a `pass` statement)

   b. Addition of two float variables

   c. Multiplication of two float variables

   d. Division of two float variables

   e. Integer division of two variables

   f. Appending one number to a list. Does this depend on the length of the list?

   g. Call to a function that does nothing (contains only a `pass` statement)

   h. Call to a function that adds two float variables

   Note: for each of these, you will need to run the operation many times in a row and divide the total elapsed time to get an accurate answer.

4. **Fibonacci Numbers.** Write a program that takes as a command line argument a single number n, then prints the first n Fibonacci numbers starting with $1, 1, 2, 3, \ldots$
   Make sure your output has no more than 75 characters per line. You may limit the output to numbers with 75 or fewer digits. Avoid recursion; it can be inefficient in Python.

5. **Sine Function.** Write a program that

   a. Prompts the user for a non-zero angle in degrees

   b. Continues to reprompt the user unless the input is acceptable

   c. Prompts the user for a number, ≤ 25, of terms to sum in a series

   d. Continues to reprompt the user unless the input is acceptable

   e. Calculates, using a function you have defined called `sind()`, the sine of the angle provided by the user. This function must sum a series to the number of terms requested, and may not use any predefined Python math functions.

   f. Calculates the sine of the angle using `math.sin()`

   g. Prints out a quantitative comparison between the two answers including the ratio and the absolute difference.

6. **Julian Day.** Download and read the Julian day handout from the course web page. Write a program that

   a. Prompts the user for a date in the form `DDMmmYYYY`. For example, `26Apr2016`.

   b. Computes and prints the Julian day number corresponding to 0 h universal time on the selected date

   c. Computes and prints the day of the week on which the selected date falls.

   Your program must use separately defined functions to

   a. Prompt the user and return his/her input string

   b. Parse the input string and return a list containing the year, month number (1–12), and day of the month

   c. Compute the Julian day number using the list returned by the function from the previous item

   d. Compute the day of the week given the Julian day number

   Use your program to find the day of the week on which you were born, and for how many days you have been alive.

   Hint: You may want to refer to the online Python documentation for the `time` module.