PROGRAMMING WORKSHEET 1

LUKE COLLINS

Running Python Programs. To run programs when using the Sublime text editor, you can just hit `Ctrl`+`B` and a terminal window should pop up with the program's output. If the program requires user input, you will need to use an external terminal. Hit `⊞`+`R`, type `cmd` in the dialogue box which comes up, and hit enter (`↵`).

When the terminal pops up, navigate to the folder containing your program using the `cd` command. For example, if the directory is

> `C:`▸`Users`▸`Bobby`▸`Desktop`▸`Computing`▸`Programs`

then type in the terminal

> `cd C:\Users\Bobby\Desktop\Computing\Programs`

and hit enter (`↵`). To check that you are in the right folder, type `dir` and hit enter (`↵`), you should see all the files in the folder listed.

To run the program `prog.py` in the folder, simply type `python prog.py` and hit enter (`↵`), here you should be able to provide user input without any issues.

EXERCISES

1. Write a program which asks the user to enter two numbers and prints out their sum.
2. Write a program which asks the user to enter a year, and check whether or not it is a leap year.[1]
3. Write a program which asks the user to input which day of the week today is (in the form of a menu, say, e.g.,

   ```
   1. Monday
   2. Tuesday
   3. Wednesday
   4. Thursday
   5. Friday
   6. Saturday
   7. Sunday
   Pick today's date (1-7): )
   ```

   and then asks the user to enter a number of days (say they enter `450`). The program should then output what that day will be, e.g.,

   ```
   In 450 days from now, it will be a Thursday.
   ```

   where the correct day of the week is computed appropriately.
   (Hint: you might need to look up the modulo (%) operator).

Date: August, 2020.

[1]Do look up what makes a year is a leap year, it's not just about being divisible by 4 as most people think. For instance, 2000 was a leap year, but 1900 wasn't.

4. (Running total). Write a program which keeps prompting the user to enter numbers to add to a running total. If the user hits enter (⏎) without a number (i.e., inputs the empty string " "), then the running total should reset to zero. If the user hits enter twice, the program should terminate. Here is an example run of this program:

```
Enter a number: 5            (user enters  5 + ⏎ )
Total: 5
Enter a number: 20           (user enters  2 + 0 + ⏎ )
Total: 25
Enter a number: -2           (user enters  - + 2 + ⏎ )
Total: 23
Enter a number:              (user enters  ⏎ )
Total: 0
Enter a number: 5            (user enters  5 + ⏎ )
Total: 5
Enter a number:              (user enters  ⏎ )
Total: 0
Enter a number:              (user enters  ⏎ , program stops)
```

5. (Fizz buzz). There is a children's game called *Fizz buzz*. The idea is to count up (say from 1 to 100), but:
   (i) if a number is divisible by 3, rather than saying the number, you say *fizz*,
   (ii) if a number is divisible by 5, rather than saying the number, you say *buzz*,
   (iii) if the number is divisible by both 3 and 5, you say *fizz buzz*.

   Write a program which plays the children's game Fizz buzz, starting from 1 and ending at 100. Thus the first few outputs on screen should be

   ```
   1, 2, fizz, 4, buzz, fizz, 7, 8, fizz, buzz, 11, fizz,
   13, 14, fizz buzz, 16, ...
   ```

6. (Collatz conjecture). A famous mathematical conjecture states that repeatedly applying the function

$$f(n) = \begin{cases} n/2 & \text{if } n \text{ is even} \\ 3n+1 & \text{if } n \text{ is odd} \end{cases}$$

to any positive integer input n will eventually reach 1. For instance, if we start with $n = 12$, then repeatedly applying f gives the chain

$$12 \xrightarrow{f} 6 \xrightarrow{f} 3 \xrightarrow{f} 10 \xrightarrow{f} 5 \xrightarrow{f} 16 \xrightarrow{f} 8 \xrightarrow{f} 4 \xrightarrow{f} 2 \xrightarrow{f} 1.$$

Implement this function in a program, and print on screen the chains for each input n between 1 and 100.

Which starting value $1 \leqslant n \leqslant 100$ produces the longest chain?

7. Write a program which finds the maximum from a given list. (You might need to search "how to find the length of a list in python"). Test it on the list `mylist = [1,7,-3,2,0,12,133,-4,12,1,2,1]`.