Homework 1

Due: October 12, 2021

In the given examples, what you type is in red and the program prints what is in black. Your program output should look exactly like the output in the examples, except that what you type won’t be in red. Also, the symbol “,” is a newline (return or enter keys).

1. (30 points) Abraham Sharp developed an infinite sum that produces $\pi$:

$$\pi = \sum_{k=0}^{\infty} \frac{2(-1)^k 3^{\frac{1}{2} - k}}{2k + 1}$$

(a) (15 points) Compute and print the resulting approximations to $\pi$ for the first 5, 10, . . ., 50 terms. Each line of your output should look like this:

After 5 terms, the approximation is 3.1426047456630846

(b) (10 points) After what term does adding extra terms stop improving the approximation? Your output, which is to be the last line of the program’s output, is to look like:

After 5 terms, the approximation 3.1426047456630846 does not change

Your numbers may differ from the examples; this is intended to show you the format of the output only.

To turn in: Please turn in the program in the file sharp.py.

2. (30 points) Write a program to determine whether a year, given as input, is a leap year. A year is a leap year if it is evenly divisible by 4, unless it is evenly divisible by 100 and not 400. So 2000 was a leap year, but 2100 and 2200 will not be. It is to ask the user to enter a year and determine whether the year is a leap year. The program then prints the result.

Your program must give an error message and exit if the user enters anything other than a positive integer.

Here is sample output. Each is from a separate run of the program.

Year> 2020
2020 is a leap year

Year> 2000
2000 is a leap year

Year> 1900
1900 is not a leap year

Year> hello
You must enter the year as a positive integer

Year> -12
You must enter the year as a positive integer

3. (10 points) Print a tic-tac-toe board. The horizontal lines for each square are to be 5 “-“ horizontally and the vertical lines for each square are to be 3 “|”‘s vertically. Where the lines intersect, put a “+“.
The output is to look like this:
4. (30 points) **Debugging:** The program *ftoc.py* should convert temperatures in Fahrenheit to temperatures in Celsius — recall the formula is:

\[ c = \frac{5}{9}(f - 32) \]

where \( c \) is the temperature as Celsius and \( f \) is the temperature in Fahrenheit. It is to print out the temperatures as decimal numbers, rounded to 2 decimal places if they are not integers. But it doesn’t work — indeed, it won’t even run to completion!

Find the bug(s) and fix it (them). Put in your comments what you changed.

*Hint:* The function to round a number \( x \) to two places is `round(x, 2)`. So, to round \( x \) to two places, say `round(x, 2)`.

**To turn in:** Please turn in the program in the file *ftoc-fixed.py*. Here are examples of the correctly working program:

Temperature in Fahrenheit: -40,
-40.0 in Fahrenheit is -40.0 in Celsius

Temperature in Fahrenheit: 99.99,
99.99 in Fahrenheit is 37.77 in Celsius

Temperature in Fahrenheit: 212.56,
212.56 in Fahrenheit is 100.31 in Celsius