CSC 110 Programming Project 2

Assignment Overview[1]

This assignment focuses on the design, implementation, and testing of a Python program to display information about the energy released by earthquakes (see below).

Assignment Specifications

For each of the following Richter scale measurements, your program will perform the appropriate calculations and display the equivalent amount of energy in joules and in tons of exploded TNT:

- 1.0
- 5.0
- 9.1 (Indonesia earthquake, 2004)
- 9.2 (Alaska earthquake, 1964)
- 9.5 (Chile earthquake, 1960; largest ever measured)

Then, your program will then prompt the user to enter a Richter scale measurement, accept a floating-point value representing that measurement, perform the appropriate calculations, and display the equivalent amount of energy in joules and in tons of exploded TNT for that user-selected value.

Assignment Notes

The Richter scale is a way to quantify the magnitude of an earthquake using a base-10 logarithmic scale. The magnitude is defined as the logarithm of the ratio of the amplitude of waves measured by a seismograph to an arbitrarily small amplitude. An earthquake that measures 5.0 on the Richter scale has a shaking amplitude 10 times larger than one that measures 4.0 and corresponds to a 31.6 times larger release of energy. [Rephrased from Wikipedia]

The energy in joules released for a particular Richter scale measurement is given by:

$$Energy = 10^{(1.5 * richter) + 4.8}$$

where *Energy* is measured in joules and *richter* is the Richter scale measurement (typically on a scale from 1-10 as a floating-point number).

One ton of exploded TNT yields 4.184×10^{9} joules. Thus, you can relate the energy released in joules to tons of exploded TNT.

[1] Based on and adopted from the CS1 Python Programming Projects Archive

You can use the website http://www.convertalot.com/earthquake_power__calculator.html to check your work.

Commentary:

- To clarify the project specifications, we have provided sample output at the end of this document.

- The `input` function is used to accept a response from the user. The function takes a string (a sequence of characters between quotes) as a prompt to display to the user. It then waits until the user types a response, terminated by the user touching the Enter key. Finally, the function returns the user's response as a string.

  If the user's response is supposed to be processed as a numeric value, the returned string must be converted into a number. When working with floating point numbers, a string is converted into a floating-point number using the `float` function. The function accepts a string as its argument and returns the floating-point number which the string represents. A typical interaction would be something like:

  ```
  num_str = input( "Please enter a number: " )
  num_float = float( num_str )
  ```

- `The print` is used to display some combination of variables, values, and strings. Each item to be displayed must be separated from the other items by a comma. By default, the items will be printed together on a single line. For example:

  ```
  val_float = 123.456
  print( "Number ", val_float," times two: ", val_float*2 )
  ```

  This command has four items to display: a string ("Number "), the value in the variable `val_float` (123.456), another string (" times two: ") and the result of evaluating an expression (246.912). It will print:

  ```
  Number 123.456 times two: 246.912
  ```

- To indicate scientific notation in Python, use the format *xey,* where *x* and *y* denote numbers (`float` or `int`). For example, `4e5` is used to indicate 4*10^5 .

- To raise a number to a power, use the ** operator. For example:

  ```
  print( "10 raised to the power 1.5 is", 10**1.5 )
  ```

  which results in:

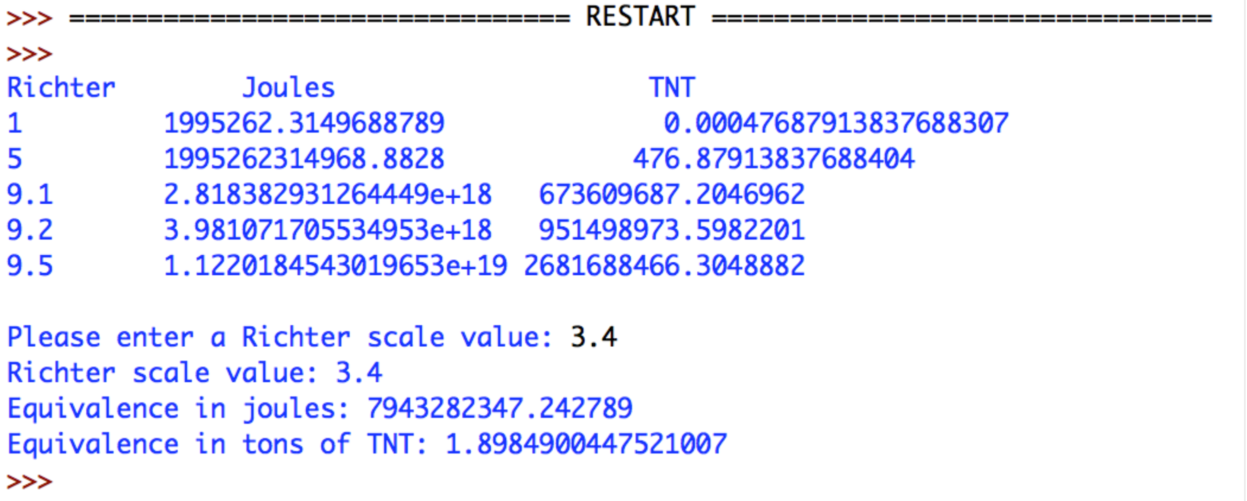
  ```
  10 raised to the power 1.5 is 31.622776601683793
  ```

- Python has many facilities to make output look nice, but we have not studied them yet. Although not required, you can do simple alignment (as in the sample output) by placing strings of spaces ("   ") of particular lengths in print statements. It will take some trial and error to get it to look better.

Suggestions

1. *Solve the problem using pencil and paper first.* You cannot write a program until you have figured out how to solve the problem. This first step is best done collaboratively with another student. (However, once the discussion turns to Python specifics and the subsequent writing of Python, you must work on your own.)
2. Use IDLE to create a new Python program (also known as a Python module):
   a. Use the required file name ("*earthenergy.py*").
   b. Make sure to save the above in a known location on your computer (ex: in *CSC110PythonCode* folder)
3. Write a simple version of the program (perhaps one which calculates the joules for an earthquake of magnitude 1 on the Richter scale).
4. Run the program and track down any errors.
5. Cycle through the steps to incrementally develop your program:
   a. Edit your program to add new capabilities.
   b. Run the program and fix any errors.
6. Submit your final program using our Canvas class assignment submission

Sample Output



```
>>> ================================ RESTART ================================
>>> 
Richter        Joules                    TNT
1         1995262.3149688789              0.00047687913837688307
5         1995262314968.8828            476.87913837688404
9.1       2.818382931264449e+18   673609687.2046962
9.2       3.981071705534953e+18   951498973.5982201
9.5       1.1220184543019653e+19 2681688466.3048882

Please enter a Richter scale value: 3.4
Richter scale value: 3.4
Equivalence in joules: 7943282347.242789
Equivalence in tons of TNT: 1.8984900447521007
>>> 
```

Implement the design

Individually write a Python program to implement your design. Start with a copy of the Python file you have that has the design in it (possibly updated with improvements you came up with) and write your Python code between the commented lines of the design. Make sure you eliminate any syntax and logic errors. Verify that it does come out with the correct answers. Your program must have a main function and call it to run the program.
Submit your solution by attaching the source code (*earthenergy.py* file).