Homework #3

Due: November 10, 2022 **Points:** 100

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.

1. (*30 points*) Write a *recursive* function to print out the depth of lists. For example, [3, [4, 5], [6, [7, [8]]]] has depth 4 because 8 is in a list that's an element in a list that is itself an element in list, which is the the element of the main list.
   Make sure your function is recursive! Also, you only need to write the function.

   Examples:

   ```
   listdepth( [ 3, [ 4, 5 ], [ 6, [ 7, [ 8 ] ] ] ])
   4

   listdepth( [ [ [ [ [ [ [ ] ] ] ] ] ] ])
   7

   listdepth([ "hello [ there ]" ])
   1

   listdepth(5)
   Should never get here -- called listdepth on non-list
   ```

   To turn in: Please call your function `listdepth`, your file *listdepth.py*, and submit it to Canvas

2. (*50 points*) The file *bp.csv*, available on Canvas, contains seven comma-separated fields. The first line contains headers describing what each field is, and the remainder of the lines contain data. For this problem the following columns are important:
   (a) Column 3, the diastolic blood pressure value;
   (b) Column 6, the systolic blood pressure value;
   (c) Column 7, the pulse pressure; and
   (d) Column 5, the patient business identifier.
   For each patient, print the following:
   (a) Number of entries for the patient;
   (b) Minimum, mean, and standard deviation of the diastolic blood pressure values;
   (c) Maximum, mean, and standard deviation of the systolic blood pressure values; and
   (d) Maximum, mean, standard deviation, and minimum of the pulse pressure values.
   You do not need to sort the output by patient ID.

   Examples (note the numbers are not as in the file; they just show you how to format the output):

   ```
   Patient ID: 139
      Entries: 7
      Diastolic blood pressure: max 83, min 61, standard deviation 9.34
      Systolic blood pressure: max 160, min 99, standard deviation 12.34
      Pulse pressure: max 72, min 60, standard deviation 3.22

   Patient ID: 27
      Entries: 19
      Diastolic blood pressure: max 100, min 70, standard deviation 10.66
      Systolic blood pressure: max 150, min 76, standard deviation 5.98
      Pulse pressure: max 81, min 62, standard deviation 6.83
   ```

   To turn in: Please call your program *pressure.py* and submit it to Canvas

3. (*20 points*) The file *atomic_weights.txt* contains lines with the atomic weight, chemical symbol, and full name of the elements. They are ordered in terms of the atomic number, with Hydrogen being 1, Helium 2, and so forth. The atomic weight is a floating point number; the atomic number is an integer. The other two fields are strings, and contains only letters (both upper and lower case).
   Write a program that read in the file and create a dictionary with four types of entries, namely: one with the atomic number as key, one with the atomic weight as key, one with the symbol as key, and one with the full element name of the key. The values are the line from the file.
   Prompt the user for a key. If the key is an integer, return the entry for that atomic number. If it is a floating point number, return the entry with the closest atomic weight. If it is a string, return the entry for that string. Continue until the user enters an end of file (control-D, written as ^D).

   Examples:

```
Enter an atomic weight, an atomic number, a symbol, or name: Mn
Manganese (Mn) has atomic number 25 and atomic weight 54.94
Enter an atomic weight, an atomic number, a symbol, or name: 5
Boron (B) has atomic number 5 and atomic weight 10.81
Enter an atomic weight, an atomic number, a symbol, or name: Oxygen
Oxygen (O) has atomic number 8 and atomic weight 16.00
Enter an atomic weight, an atomic number, a symbol, or name: 5.10
The closest: Helium (He) has atomic number 82 and atomic weight 4.00
Enter an atomic weight, an atomic number, a symbol, or name: Salt
Salt is not in the database
Enter an atomic weight, an atomic number, a symbol, or name: ^D
Goodbye!
```