PYTHON EXAM 3

- This is an examination. It must be your work and your work alone.
- No exchange of information with another human entity in any form is acceptable.
- It is ok to read google documents, however exchanges of information about the exam via the internet are forbidden.
- It is ok to use class material, notes, your programs, your labs, and any other notes you have written for class.
- It is ok to use a textbook.

These instructions are general for all the scripts that you will write in Python.

```
Instructions for Python Exam submission:
# jhed_ID:date:filename

#a
Python code

#b
Python code
```

```
General Grading
Full points for each correct answer
Partial points for each section returning error messages
```

After you download your Exam3-Spring2022.zip from Canvas, unzip the file if necessary, and a directory called **Exam3-Spring2022** will be created. Open Python and change into Exam3-Spring2022 and start your work in that directory.

Policy for cheating: sharing code on the exam is unacceptable and will earn you a 0 and take you straight to the ethics board. Do not share any study documents either!

We give partial points in the case of syntax errors.

1. (35) The provided file **freq.dat** contains frequencies (numbers) in the visible spectrum from 400 to 800 THz.

   In this exercise, we will focus on frequencies associated with the color *violet*, *blue* and *green,* and we name the rest of the frequencies *out of range*.

violet	668 < frequency <= 789 THz
blue	606 < frequency <= 668 THz
green	526 < frequency <=606 THz
out of range	frequency <= 526 or frequency > 789 THz

   Write a script called **freq_color.py** and in it:

   Write the honor sentence in comment line

   # I promise not to communicate with another human being in any way about this exam.

   a. (1) Import Numpy and pandas

   b. (3) Read in the file **freq.dat** by using a NumPy function we did in class.

   c. (5) What is the total number of frequencies stored in the data set? Do not use loops. Display the formatted result to screen.

```
There are 1e+03 frequency values
```

   d. (12) Make this DataFrame, and called it T.

```
              Color  Count
668-789 THz  violet    307
606-668 THz    blue    165
526-606 THz   green    211
```

   The Count columns contains the number of frequencies for each color (violet, blue, or green). Use Boolean arrays, and the sum function.

   e. (8) What is the color with the largest count? Use a function from NumPy or pandas, and access values in the DataFrame T. Display the result to screen, which should be:

```
violet
```

2. (44) The depth of snow (in inches) has been measured in a very cold location every week since the snow began accumulating. At this point, the season has changed, and it is getting warmer, so the pile of snow is beginning to recede, but it hasn't all gone away yet. The depths have been recorded every week and stored in the provided file called **snowdata.csv**.

   View the contents of the file. In the data file:
   the 1st field contains the week number
   the 2nd field the depth of snow in inches

   Write a script called **snowgone.py** to predict in which week the snow will be totally gone by fitting a quadratic curve through the data points. This will be called the "snow gone week number".

   a. Import NumPym, pandas and Matplotlib

   b. (6) Use pandas to read in the data file **snowdata.csv**, and store in variable C.
      See the content of the file before reading it in. The first line is a comment line, and the second line reports the names of the fields, which should name the columns. The rows should contain numeric indices. Display the DataFrame to screen.

```
    week  depth
0      1      8
1      2     20
2      3     31
3      4     42
4      5     55
5      6     65
6      7     77
7      8     88
8      9     95
9     10     97
10    11     89
11    12     72
12    13     68
13    14     53
14    15     44
```

   c. (2) Store the 1st field in variable **week** and the 2nd field in variable **depth**.

   d. (5) Fit the data (i.e., **depth** (y axis) vs **week** (x axis)), by using the polynomial of degree 2. Display to screen the result of the fitting model.

   e. (4) Calculate the R^2 value. Print the following sentence to screen and format the R^2 value to 2 decimal digits:

```
The R-squared value is 0.94
```

f. (6) Use the fitting model to predict the depth of the snow from the last recoded week to week 25.
For this make a 1D array called **predweek**, which contains numbers from the last recorded week to week 25 in steps of 0.5.
Do not hardcode the last recorded week.
If you do not know how to obtain the last recorded week, for a loss of 2 points you can hardcode the number 15.

Use array **predweek** and the fitting model to predict the corresponding depth of the snow, and store results in variable **preddepth**. Do not hardcode the fitting model, but use the variable generated in the previous point.

g. (6) Use **predweek** and **preddepth** to find the snow gone week.
The snow gone week is the number in **predweek** which corresponds to the last positive value in **preddepth**.

Store the gone week in variable **goneweek**, and the corresponding depth in variable **gonedepth.** Use Boolean vector.

h. (3) Use variable **goneweek** to print to screen:

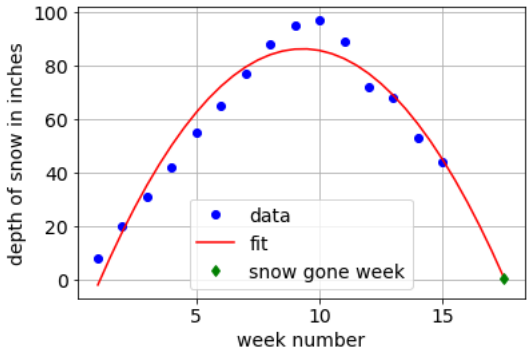
```
The snow gone week is 17.5
```

i. (11) Make the figure reported below:
- Plot the data, **depth** vs **week** with a blue circle.

- Plot the fitting model with a red solid line.
  For this make two arrays **linex** and **liney.**
  linex should contain numbers from 1 to the goneweek, in step of 0.5
  liney, should be the corresponding depth obtained by using the fitting model.

- Plot the snow gone week, and corresponding depth by using variables **goneweek** and **gonedepth**. Use a greed diamond.

- Label the axis, make the legend and the grid as below.
  Set the font size to 14 us rcParams.



3. (30) The eccentricity of an ellipse is defined as:

$$e = \sqrt{1 - \left(\frac{b}{a}\right)^2}$$

where a is the semimajor axis and b is the semiminor axis of the ellipse., i.e., here a should be greater than b, i.e., a > b. Make a script, **ecce.py** and in it do the following.

a. (1) Import Numpy

b. (8) Define a function, called **eccentricity**. The function takes two arguments, a and b, and calculates and returns the eccentricity value. Use Numpy functions

c. (3) Generate a 1D array **a_v** containing 10 integer random numbers in range 20-30, representing semimajor axis values.

d. (3) Generate a 1D array **b_v**, containing 10 integer random numbers in range 2-10, representing semi-minor axis values.

e. (4) Call function **eccentricity** to calculate the eccentricity for the values stored in arrays **a_v** and **b_v** and store result in variable **eccen_v**. Do not use loops.

f. (11) Use Boolean vectors and find the eccentricity values greater then 0.95, and corresponding values in a_v and v_b. Print the results formatted: Keep in mind that these are random numbers so the output could be different.

```
0.99 30 4
0.99 28 3
0.97 26 6
0.98 20 4
0.95 30 9
```

EC(2) Write a script **ECs22.py** that produces the following output. The script should iterate from 1 to 9 to produce the expressions on the left, perform the specified operation to get the results shown on the right, and print exactly in the format shown here.

```
1 x 8 + 1 = 9
12 x 8 + 2 = 98
123 x 8 + 3 = 987
1234 x 8 + 4 = 9876
12345 x 8 + 5 = 98765
123456 x 8 + 6 = 987654
1234567 x 8 + 7 = 9876543
12345678 x 8 + 8 = 98765432
123456789 x 8 + 9 = 987654321
```

Upload the following files to Gradescope Exam3:
freq_color.py
ecce.py
snowgone.py
ECs22.py
DO NOT SUBMIT ANY ADDITIONAL FILES