ENGR 102 – Fall 2022
Lab: Topic 11 (individual)
34 points = 100 % Canvas

Deliverables:
There are several deliverables for this individual assignment. Please submit the following files to zyBooks:

- `barcode_checker.py – [11]  11.11  [8 auto grading +3 manual]`
- `count_coins.py  [11]  11.12  [8 auto grading +3 manual]`
- `weather_data.py  [12]  11.13  [12 automated]`

These activities are meant to help give you practice reading and writing files, as well as processing larger amounts of data, which is one of the common tasks that computer programs are written for.

Activity #1: Barcode checker – individual
A barcode is valid if the digits satisfy a certain constraint. For example, take the 13-digit barcode `1877455846014` and split the first 12 digits into two groups: `(1,7,4,5,4,0)` and `(8,7,5,8,6,1)`. The first group contains every other digit starting with the first, and the second group contains every other digit starting with the second. Take the sum of the digits in the second group, and multiply it by 3. Add to that the sum of the digits in the first group. Subtract the last digit of that number from 10, and it should match the last digit of the barcode.

Example math using barcode `1877455846014`:
Sum of first group = 1 + 7 + 4 + 5 + 4 + 0 = **21**
Sum of second group = 8 + 7 + 5 + 8 + 6 + 1 = **35**
Multiply second group by 3 = **35** x 3 = 105
Add first group = 105 + **21** = 12**6**
Use the last digit and subtract from ten = 10 - **6** = 4
4 is the last digit in the barcode, so it is valid

Write a program named barcode_checker.py that takes as input a filename that contains many 13-digit barcodes. Have your program read the file, determine whether each barcode is valid, and write the valid barcodes to a new file named valid_barcodes.txt. Have your program output the total number of valid barcodes found using the example output below. You do not have to submit your valid_barcodes.txt file to zyBooks.

Example output using **`barcodes.txt`**:
```
Enter the name of the file: barcodes.txt
There are ?? valid barcodes
```

Activity #2: Counting coins – individual
While digging through a box of very old handheld game consoles, you find one that piques your interest. Wondering how it works, you access the source code and find a text file named game.txt full of instructions, one per line. Each instruction consists of an operation (coin, jump, or none) and a signed number (like +25 or -3). You quickly figure out that coin increases or decreases a value that stores the number of coins earned by the player, jump will jump to a new instruction relative to itself, and none does absolutely nothing. After executing a coin or none operation, the instruction immediately below is executed next. However, jump +2 would continue to the instruction 2 lines below it, and jump -5 causes the instruction 5 lines above to be

executed next. The program ends when it attempts to execute an instruction immediately after the last instruction in the file.
Write a program named count_coins.py that opens the game file (game.txt), executes the instructions, and creates a new file named coins.txt that contains only the numbers of coins gained or lost in the order the program is executed. Have your program output the total number of coins earned using the example output below. You do not have to submit your coins.txt file to zyBooks.

Example output:
```
Total coins collected: ???
```

Example coins.txt file created by your program:
```
29
...
-87
4
8
```

Activity #3: Weather data – individual
On Canvas, there is a CSV file posted with this assignment named WeatherDataCLL.csv that contains weather data from Easterwood Airport (in College Station) for 3 years. The data was taken from the National Oceanic and Atmospheric Administration's National Centers for Environmental Information[1]. You can view the data by opening the file in any text or spreadsheet editor (e.g. Notepad++, Excel) or in Spyder. The first line of the file contains the column headers explaining what each column is.

Download the file and write a program named weather_data.py that does the following:

1. Open the CSV file for reading
2. Read the CSV file and compute
    a. the maximum temperature seen over the 3-year period
    b. the minimum temperature seen over the 3-year period
    c. the average daily precipitation over the 3-year period (use 3 decimal places)
3. Output the results to the console using the format below
4. Perform the following three data analysis exercises and output the results to the console. Take as input from the user a month and year, then for that month,
    a. Calculate the mean of the maximum temperatures (use 1 decimal place)
    b. Calculate the mean daily wind speed (use 2 decimal places)
    c. Calculate the percentage of days with non-zero precipitation (use 1 decimal place)

Example Output (using inputs **July**, **2021**, but with made-up numbers):
```
3-year maximum temperature: 101 F
3-year minimum temperature: 7 F
3-year average precipitation: 0.123 inches

Please enter a month: July
Please enter a year: 2021

For July 2021:
Mean maximum daily temperature: 92.5 F
```

[1] NOAA's NCEI, https://www.ncdc.noaa.gov/

```
Mean daily wind speed: 7.54 mph
Percentage of days with precipitation: 23.1%
```