HW10-PART1 (30 POINTS)

These instructions are applicable for all scripts that you will write in python3.

```
Instructions for Python3 Homework submission

#class header

#Q1
var1=124

#Q2
var2=3+var1
```

```
HW10-part1 is due Sunday April 14th 11:59pm

If you submit it by Wed Aril 10th 11:59pm you will get 3 extra credit points
```

Policy: Work on the HW problems on your own. It is not allowed to share code in any way. The use of chatGPT is not allowed. Using the internet to find out how to solve problems is not allowed.

General Grading rules we will follow:

- We do not give points for instructions but take points off for not following them.
- We take all points off for a question if your code has syntax errors (comment out the code for partial credit).
- We grade the code, not the output of the code. Even if the output is correct, hard-cording, redundant logic, code that has no purpose in the program will not earn full credits.
- We will take points off if you use out-of-class material.
- We will take points off if you do not follow directions when a question specifies to use a certain command or write the code in a certain way.

HW10-part1 (30 points)

1. (15) Population data for different cities is stored in a dictionary. We want to analyze this data using pandas Series, where each city's name serves as the index label, and their corresponding population serves as the value in the Series.

In the provided **ex10-cities.py** this dictionary has been defined:

```
D = {'New York': 8500000,
'Los Angeles': 4000000,
'Chicago': 2700000,
'Houston': 2300000,
'Phoenix': 1600000,
'Philadelphia': 1500000,
'San Antonio': 1400000,
'San Diego': 1300000,
'Dallas': 1200000}
```

Edit the provided file and do the following:

Keep in mind we take all points off for syntax errors.

Q1 (5) Create a pandas Series out of dictionary D, containing the first 6 cities' names as index labels and corresponding population counts as values. Assume you do not know what the first 6 cites are in the dictionary. Use the index parameter in pd.Series().

Print the Series to screen.

```
New York        8500000
Los Angeles     4000000
Chicago         2700000
Houston         2300000
Phoenix         1600000
Philadelphia    1500000
dtype: int64
```

Q2 (2) Print the index labels as a 1D array to the screen:

```
['New York' 'Los Angeles' 'Chicago' 'Houston' 'Phoenix' 'Philadelphia']
```

Q3 (1) Slice the Series to obtain population data for the first 3 cities, and print it to screen:

```
New York       8500000
Los Angeles    4000000
Chicago        2700000
dtype: int64
```

Q4 (2) Select population data for cities named Los Angeles, Houston, and Phoenix, and print it to screen.

```
Los Angeles    4000000
Houston        2300000
Phoenix        1600000
dtype: int64
```

Q5 (2) Select population data greater than 2 million and print it to screen.

```
New York       8500000
Los Angeles    4000000
Chicago        2700000
Houston        2300000
dtype: int64
```

Q6 (3) Count the number of cities with populations greater than 5 million, and print to screen:

```
Number of cities with populations greater than 3 million is: 2
```

2. (15) Suppose we have collected data on various chemical compounds, including their molecular weights, boiling points, and whether they are organic or inorganic. A pandas DataFrame provides a convenient data structure for organizing and analyzing this kind of information, allowing easy manipulation and exploration of the data.

In the provided script **ex10-chemistry.py** this dictionary has been defined:

```
D={'Compound': ['Water', 'Ethanol', 'Carbon dioxide', 'Ammonia', 'Methane'],
'Molecular Weight (g/mol)': [18.015, 46.07, 44.01, 17.031, 16.043],
'Boiling Point (C)': [100, 78.37, -78.5, -33.34, -161.5],
'Organic': [False, True, False, False, True]}
```

Q1 (4) Given the following dictionary containing data on chemical compounds, create and print this pandas DataFrame. Notice you should rename the row labels.

```
                Molecular Weight (g/mol)  Boiling Point (C)   Organic
Compound
Water                             18.015             100.00    False
Ethanol                           46.070              78.37     True
Carbon dioxide                    44.010             -78.50    False
Ammonia                           17.031             -33.34    False
Methane                           16.043            -161.50     True
```

Q2 (1) Display the 1D array of row labels, and the 1D array of column labels.

```
['Water' 'Ethanol' 'Carbon dioxide' 'Ammonia' 'Methane']
['Molecular Weight (g/mol)' 'Boiling Point (C)' 'Organic']
```

Q3 (1) Rename the columns so that the DataFrame becomes:

```
                Molecular Weight  Boiling Point  Organic
Compound
Water                     18.015         100.00    False
Ethanol                   46.070          78.37     True
Carbon dioxide            44.010         -78.50    False
Ammonia                   17.031         -33.34    False
Methane                   16.043        -161.50     True
```

You can hardcode a list of the new columns and use it to relabel the columns. Print the DataFrame to screen.

Q4 (1) Extract and print data for the compounds Ethanol, Ammonia, and Methane.

```
          Molecular Weight  Boiling Point  Organic
Compound
Ethanol             46.070          78.37     True
Ammonia             17.031         -33.34    False
Methane             16.043        -161.50     True
```

Q5 (4) Extract and print data for organic compounds (Organic is True) with molecular weights greater than 30 g/mol.

```
          Molecular Weight  Boiling Point  Organic
Compound
Ethanol              46.07          78.37     True
```

Q6 (4) Extract and print data for non-organic compounds and display to screen their boiling point and molecular weight.

```
                Boiling Point  Molecular Weight
Compound
Water                  100.00            18.015
Carbon dioxide         -78.50            44.010
Ammonia                -33.34            17.031
```

```
Upload to Gradescope HW10-part1:
ex10-cities.py
ex10-chemistry.py
```