****Assignment 5****

Note: Friday's lecture included a program that was similar to this assignment: [phoneDir.py](https://canvas.ucdavis.edu/courses/253898/files/4530763/download?verifier=e9vnNtjtabm5pKrev9qjAt6Oa7Ei83jrKTO4RSDx) it uses the files [areaCodes.txt](https://canvas.ucdavis.edu/courses/253898/files/4502298/download?verifier=l05Hl3UMuV7J0HLcaAyDD6vNwt0DRQSlzCckShP3) and [phoneDir.txt](https://canvas.ucdavis.edu/courses/253898/files/4512862/download?verifier=LfpcOKMT7v73RkYjvSITi5ONhBpRcxeNtspezciX) to make a dictionary mapping area codes to states and filter a list of phone numbers by state. Understanding how that program works will help you with this assignment.

****Drinking Water Filtered by Population****

At the consulting company that hires you with your double major in Spanish and Economics, you are helping to develop a report for a client on how to improve access to safe drinking water in developing countries. To get started, you decide to figure out which countries are doing a good job of improving access to safe drinking water.

Fortunately, you find this file [drinkingWater.csv](https://canvas.ucdavis.edu/courses/253898/files/4437258/download?verifier=uksKUExZXxDSNDbngm52wTpxB1cusWoBIez5YIM7). It's part of the dataset downloaded from the bottom of this page from the [Guardian (Links to an external site.)Links to an external site.](http://www.guardian.co.uk/global-development/poverty-matters/2013/jan/23/hunger-malnutrition-key-datasets" \t "https://canvas.ucdavis.edu/courses/253898/assignments/_blank), a newspaper in the UK that publishes a lot of data files. It contains the percentage of people in each country who had access to an "improved water source" in the years 1990-2010 (they say "Access to an improved water source refers to the percentage of the population with reasonable access to an adequate amount of water from an improved source, such as a household connection, public standpipe, borehole, protected well or spring, and rainwater collection. Unimproved sources include vendors, tanker trucks, and unprotected wells and springs. Reasonable access is defined as the availability of at least 20 liters a person a day from a source within one kilometer of the dwelling.")

But some of these countries are pretty small, and they are probably not relevant to the client who wants the report. So you decide to consider only countries that have a population greater than 500,000 (Davis is already over 50,000, and San Jose is a million; so we are talking about excluding really small countries).

To get the populations, we'll use the file [world\_population\_2017.tsv](https://canvas.ucdavis.edu/courses/253898/files/4437257/download?verifier=I8uEktl8uRwumj73O8prKaObjY2ULeJnAjfk35CZ) obtained from the [CIA World Factbook (Links to an external site.)Links to an external site.](https://www.cia.gov/library/publications/the-world-factbook/rankorder/rankorderguide.html" \t "https://canvas.ucdavis.edu/courses/253898/assignments/_blank).

To get you started, we have provided a file [HW5.py](https://canvas.ucdavis.edu/courses/253898/files/4437356/download?verifier=XB0ZEA6TFpyVHNv84leIMnkNUVcOUYjAt4XZwtS6) with just the main function, and stubs for the two functions that do the real work. Apart from function calls, the code should go into the two functions makeDictionary and readDWdata.

The makeDictionary function should read in population.tsv, and build a dictionary in which the keys are countries and the values are populations.

The readDWdata function should read in the file drinkingWater.csv, and print out the difference for any large enough country where the percentage of people with access to safe water changed between 1990 and 2010.

****Part 1 (40 points)****

Read in [world\_population\_2017.tsv](https://canvas.ucdavis.edu/courses/253898/files/4437257/download?verifier=I8uEktl8uRwumj73O8prKaObjY2ULeJnAjfk35CZ) and create dictionary.

Start out by filling in the makePopDictionary function definition so that it reads the input file world\_population\_2017.tsv. You are already familiar with reading comma separated files from the last assignment. In the tsv file the row number, country name, and population are separated by the tab character "\t", instead of a comma.

You are interested in extracting the country name and population as variables. You will need to convert population from a string to an integer. The string method replace() can be used to remove the commas from the numbers by replacing them with the empty string.

Add a temporary print statement to check that you correctly extracted the data from the file; e.g. print(country,population).

Test your code by calling makePopDictionary from main and running your program.

You should see the following at the end when it is done. The values will be printed in the same order as in the file, but separated by a space instead of a tab.

China 1379302771  
India 1281935911  
United States 326625791  
Indonesia 260580739  
Brazil 207353391

When you are satisfied it works, you can now populate the dictionary and return it from the function.

Upload your file to Kodethon as ****HW5pt1.py****

****Part 2 (40 pts):****

Read the drinkingWater.csv file and compute access

Start by removing the print statement in makePopDictionary and adding a call to readDWdata in main(). Your test program will now be printing out

In the readDWdata function, write the code for reading in the drinkingWater.csv file and extracting the country name and the percentages for 1990 and 2010. If a country does not have percentages for both year then you will skip it. Next compute how access has changed by subtracting the percentage in 1990 from the percentage in 2010. Print out the name of the country, the two percentages, and the difference.

The first five lines of your output should look like:

World 76  88 12  
Developing countries 70  86 16  
Africa 56  66 10  
North Africa 87  92 5  
Algeria 94  83 -11

Upload your file to Kodethon as ****HW5pt2.py****

****Part 3 (20 pts)****

Now we are only interested in printing the values only for countries that match keys in the population dictionary and have populations over 500,000. You do not have to worry about country names that don't match exactly, such as United States of America vs United States. Your program can mistakenly assume that these are different countries.

Add a first line that describes the column contents (see below). You do not have to write the output to a file for this assignment, it will be sufficient to print it to the screen.

Now, the first five lines of your output should look like:

Country 1990 2010 Change  
Algeria 94  83 -11  
Egypt 93  99 6  
Morocco 73  83 10  
Angola 42  51 9

Upload your file to Kodethon as ****HW5pt3.py****