***BE 1600***

***Introduction to***

***Programming and Computation***

***Python***

**Extra Credit**

40 points

+ 10 extra points if you solve the last two questions

**Due 11/29/2021 (11:45 A.M.)**

Assignment Objectives:

* To manipulate string data type using string operators, functions, and methods
* To write simple cryptographic algorithms
* To use Python lists as a means of storing data
* To use Python dictionaries to store associative data
* To use text and cvs files to store large data sets
* To analyze real data
* Practice with file I/O.
* Process and analyze real data

*Solution for this assignment will not be posted on Canvas; however, the solution of any of the assignment problems can be discussed in the class upon request of a student.*

All assignments must be submitted by the Canvas. **No email or hard copy** is accepted. You must follow the following format:

1. For non-programming questions, use a word file to type your answers. Don’t use the text box on the Canvas to answer the questions or to write comments, we will not read it. State your answer clearly.
2. For programming questions, include only the source file of each programming problem.
3. Submit your files to the Canvas. You must submit your files on time; otherwise, you will receive zero.
4. Use “Add Another File” feature on Canvas to upload each additional file; do not upload the files as a compressed folder.
5. You can upload your files as many times as you like. Only the last attempt counts because the last files you uploaded are the only files your instructor will see.
6. There will be several modules on the Canvas. You need to upload your files using the correct module on the Canvas.
7. Name each file: *Assignment (assignment number)* for the word file [e.g. Assignment 02] and *Assignment (assignment number) \_ (Question number)* for each programming question [e.g. Assignment 02\_Q03].
8. To upload your file(s):

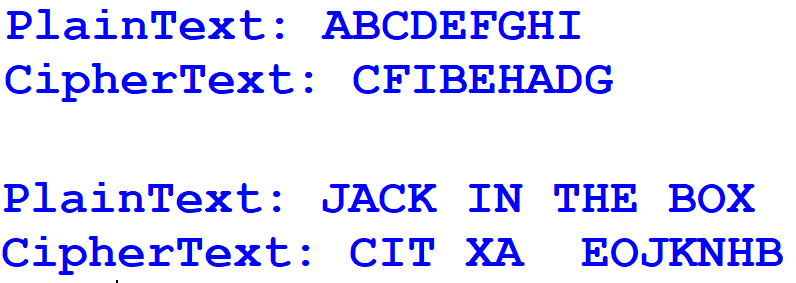
* In Course Navigation, click the ASSIGNMENTS module.
* Click the title of the assignment.
* Click the **Submit** Assignment button.
* Add **File**. ...
* Add Another **File**. ...
* **Submit** Assignment. ...
* View **Submission**.

*It is your responsibility to make sure that each file is uploaded correctly. If you uploaded a wrong file, you receive zero; files will not be accepted after due date even if you have a prove that the file is created before the due date.*

***Make sure you review the Cheating & Plagiarism policy on Canvas.***

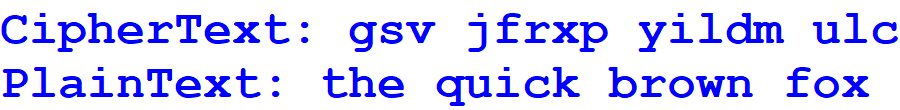
Use Python Script Window and write a program for Q.1. and Q.4. Submit four files (four .py files) to Canvas by the due date.

1. (5 points) The transposition cipher can be generalized to any number of rails. Write a function to implement a three-rail fence cipher that takes every third character and puts it on one of the three rails. Test your algorithm with two sample input as shown below.



1. (5 points) Write the substitutionDecrypt method. Test your algorithm as shown below. Your code should work for any key.

Sample output using testKey1 (‘zyxwvutsrqponmlkjihgfedcba’) and cipherText message ('gsv jfrxp yildm ulc') from substitution.py file (see source file under ch. 03 link on Canvas)



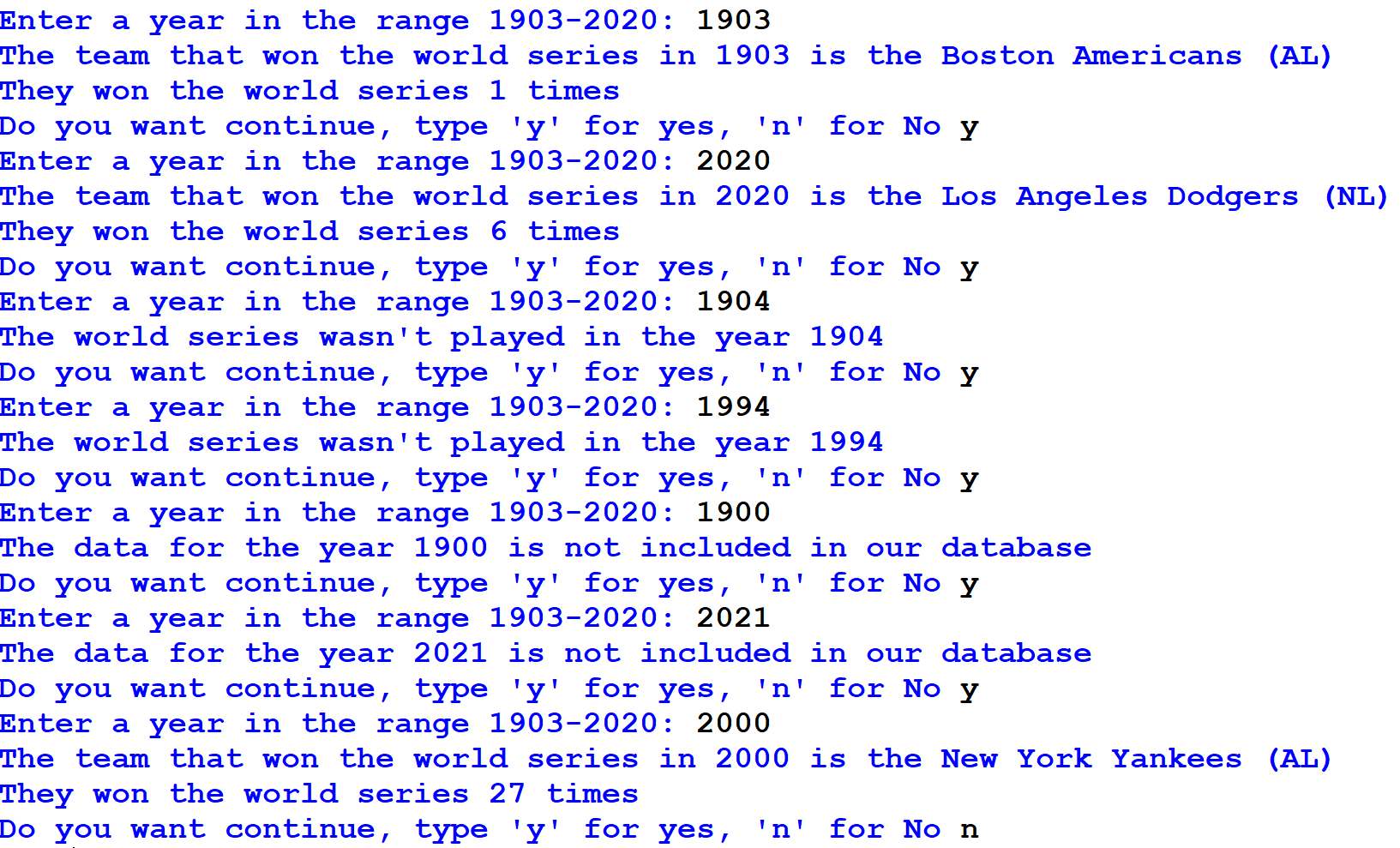
1. (10 points) The text file WorldSeries.txt contains a chronological list of the World Series’ winning teams from 1903 through 2020. The first line in the file is the name of the team that won in 1903, and the last line is the name of the team that won in 2020. (Note that the World Series was not played in 1904 or 1994. There are entries in the file indicating this.)

Write a program that reads this file and creates a dictionary in which the keys are the names of the teams and each key’s associated value is the number of times the team has won the World Series. The program should also create a dictionary in which the keys are the years and each key’s associated value is the name of the team that won that year.

The program should prompt the user for a year in the range of 1903 through 2020. It

should then display the name of the team that won the World Series that year and the number of times that team has won the World Series. Allow the user to enter more than one year and check of the World Series was not played or year is not in database.

Sample output:



1. (20 points) The Centers for Disease Control (CDC) keeps data on what it calls “Winnable Battle Risk Factors and Health Indicators”. These are negative behaviors or incidents that could possibly be avoided by changes in lifestyle. In this problem you will examine some CDC data to discover the states with the best and worst records in regards to a few of these risk factors and health indicators.

***Problem Specifications***

The file “riskfactors.csv” lists data on 20 different risk factors and health indicators for each state. The data is in “comma separated value” format (csv), which means that each entry is separated from the others by a comma. Examining the file in a text editor or a spreadsheet program or both should help you understand the format. We grabbed this file from the CDC web page and have placed a copy in the project directory for you.

Since analyzing 20 different indicators can be a bit confusing, we will only look at five: *Heart Disease Death Rate*, *Motor Vehicle Death Rate*, *Teen Birth Rate*, *Adult Smoking*, and *Adult Obesity*. Your program will read in the data from the csv file and find the states with the best and worst record for each of these indicators (largest and smallest values). It will produce a file called “best\_and\_worst.txt” which lists the states that have the highest and lowest value for each of the indicators, along with their values.

Your program must be general enough to work with a similar file with states as rows and with the same column headers. That is, if the CDC puts out a new file with different values in the cells (e.g. maybe new research changed some values), your program will work correctly.

***Additional Requirements***

1. You must create and use at least 2 meaningful functions (your choice).
2. Your program must format the file into columns (see below). You don’t have to match our formatting exactly, but columns should line up and it should be readable. Use string formatting.

**Output of the file best\_and\_worst.txt :**

**Indicator : Min Max**

**---------------------------------------------------------------------------------------**

**Heart Disease Death Rate (2007): Minnesota 129.8 Mississippi 266.5**

**Motor Vehicle Death Rate (2009): District of Columbia 4.8 Wyoming 24.6**

**Teen Birth Rate (2009) : New Hampshire 16.4 Mississippi 64.2**

**Adult Smoking (2010) : Utah 9.1 West Virginia 26.8**

**Adult Obesity (2010) : Colorado 21.4 Mississippi 34.5**

***Hints***

1. Not all lines of the file contain state data. In particular, the first few lines are irrelevant.
2. Don’t forget to convert strings to numbers where appropriate.
3. Watch out for the fact that some data has a percent sign.
4. To open a file for output, remember:
   1. Open the file with the 'w' mode string.
   2. You can only write strings to a file, so you must convert each output to a string before you write them.
   3. Also, remember that if you want a separate line to occur in your output file, you must specifically output the carriage return/line feed string "\n".
5. Don’t forget to close your file—otherwise the string might not get written. If you find that your file has nothing in it or is missing information, forgetting to close the file is a likely reason.
6. Depending on how you design your program you may find some of the following list functions and methods useful: sort, min, and max.