Honors Computer Science Python
Mr. Clausen
Program 4A, 4B, 4C

PROGRAM 4A Full Names (25 points)

This program should ask the user for their full name: first name, one space, middle name, one space, and last name, then press return. Your job is to separate this full name into three separate variables: one for the first name, one for the middle name, and a third for the last name. Display these separate names on the screen. Also display the LastName, a comma, a space, and the FirstName. You will need to use Python's string manipulation features. Here's a hint, the spaces tell you where the first, middle and last names can be found.

1) Use a `DocString` at the beginning of the program for your comments.

2) Initialize all of the variables that are to be used in this program.

3) Use print statements to display your name and period output just like those used for program 1A.

4) Ask the user for the full name.

5) Perform all of the calculations (string processing).

6) Make sure that you use descriptive identifiers for all of your variables.

7) Echo the full name, display the first name, middle name, last name, and the LastName, FirstName.

8) Save your program as LastNameFirstNameP4A.py.

Program 4B Adlibbed MadLibs (40 points)

Write a program that asks the user to enter 3 different nouns, verbs, adverbs, and adjectives.
Make your program user friendly by prompting them for these values. Use one line comments to separate this program into its parts: input, and "calculations and output" which will be merged into one section. (Parts of Speech: http://www.englishclub.com/grammar/parts-of-speech_1.htm)

1) Use a `DocString` at the beginning of the program for your comments.

2) Initialize all of the variables that are to be used in this program.

3) Use print statements to display your name and period output just like those used for program 1A.

4) Ask the user for the values of 3 nouns, verbs, adverbs, and adjectives.

5) Calculate a random number from 1 to 3.

6) Make sure that you use descriptive identifiers for all of your variables.

7) For the output section, use the random number to select one of three possible stories to fit the 3 nouns, verbs, adverbs, and adjectives into. (Use "if, elif, else statements for this.)

8) Save your program as LastNameFirstNameP4B.py.

Program 4C Word Jumble (50 points)

Write a program that reads a text file of vocabulary words (VocabList.txt). Select one of the words at random, and turn the word into a "word jumble" by scrambling all the letters of the word. Allow the user as many guesses as there are letters in the word. Keep track of the number of guesses and give the user feedback as to how many guesses remain and/or if they guessed the word correctly. If they guess the word, tell them how many guesses it took.
Make your program user friendly by prompting them for their guess. Use one line comments to separate this program into its parts: input, and "calculations and output" which will be merged into one section.

1) Use a `DocString` at the beginning of the program for your comments.

2) Initialize all of the variables that are to be used in this program.

3) Use print statements to display your name and period output just like those used for program 1A.

4) Read the contents of a text file named "VocabList.txt" from the current working directory.

5) Choose one of the words at random.

6) Scramble the letters of the selected word into a "word jumble".

7) Make sure that you use descriptive identifiers for all of your variables.

8) Ask the user to enter their guess, check this against the word chosen from the text file and tell the user whether they guessed the word correctly or not. If incorrect, tell the user how many guesses are left. If the user guesses the word, tell them how many guesses it took. Limit the number of guesses to the number of letters in the word.

9) Save your program as LastNameFirstNameP4C.py.

Program 4D Play Don't Pay, You Won't Win Anyway (40 points)

Write a program that asks the user which lottery game they wish to play and how many "quick pick" tickets they want. Display the numbers on the monitor screen and save them to a text file named, "QuickPicks.txt". Below is a list of games, the number choices and the odds of winning.
Mega Millions: Pick five lucky numbers from 1 to 75 and one MEGA number from 1 to 15: Odds 1 in 258.9 mil
Powerball: Pick five lucky numbers from 1 to 59 and one POWERBALL number from 1 to 35: Odds 1 in 175,223,510
SuperLotto Plus: Pick five lucky numbers from 1 to 47 and one MEGA number from 1 to 27: Odds 1 in 41,416,353
Make your program user friendly by prompting them for their menu choice. Use one line comments to separate this program into its parts: input, and "calculations and output" which will be merged into one section.

1) Use a `DocString` at the beginning of the program for your comments.

2) Initialize all of the variables that are to be used in this program.

3) Use print statements to display your name and period output just like those used for program 1A.

4) Display a menu for the user to select which game they wish to play. Ask them how many Quick Pick tickets they want.

5) Generate the five numbers plus the sixth number and turn this into a string for each Quick Pick they requested.

6) Display the numbers on the screen and save them to a text file named "QuickPicks.txt".

7) Make sure that you use descriptive identifiers for all of your variables.

8) Save your program as LastNameFirstNameP4D.py.

Program 4E Cryptic Code (40 points)

Write a program that reads a text file named "OriginalMessage.txt", encrypts it using any encryption algorithm you wish to use. Save the encrypted file in a text file named "EncryptedMessage.txt". Next open the encrypted file and decrypt it and save it in a file named "DecryptedMessage.txt". Use a menu to let the user choose each of these actions.
Make your program user friendly by prompting them for their choice. Use one line comments to separate this program into its parts: input, and "calculations and output" which will be merged into one section.

1) Use a `DocString` at the beginning of the program for your comments.

2) Initialize all of the variables that are to be used in this program.

3) Use print statements to display your name and period output just like those used for program 1A.

4) Display a menu so the user can choose each of the actions for this program.

5) Open the original message and encrypt it.

6) Save the encrypted file as a text file.

7) Read the encrypted file, decrypt it and save it as a text file.

8) Make sure that you use descriptive identifiers for all of your variables.

9) Save your program as LastNameFirstNameP4E.py.