CSCI 1300 - Introduction to Computer Programming
Instructor: Knox
Assignment 9
Due Sunday, April 23rd 5:00pm

For this assignment, you need to submit one file called Assignment9_LastName.py, where LastName is your last name. Please include comments in your code that explain what your code is doing. The comments should also include your name, recitation TA, and the assignment number. Each function should be commented with the functions purpose and description of the parameters.

For each of the following problems, write the named function to solve the problem. You should place all the functions and any support functions within the single file. Make sure to place your testing code within a main function or within the "`if __name__ == '__main__':`" conditional as shown in class.

1. The U.S. Census provides information about the current U.S. population as well as approximate rates of change. Three rates of change are provided:

   - There is a birth every 7 seconds
   - There is a death every 13 seconds
   - There is a new immigrant every 35 seconds

   Using those three rates of change, create a function (**`CalcNewPopulation`**) to calculate the next year's (365 days) population given a current population. The function should return the new population as an integer value.

2. Each day has 86,400 seconds (24*60*60). Write a function (**`BreakoutTime`**) that takes a number of seconds and prints the number of days, hours, minutes, and seconds in a list.
   For example, 70,000 seconds is 0 days, 19 hours, 26 minutes, and 40 seconds. Your function would print :
   ```
   Days: 0
   Hours: 19
   Minutes: 26
   Seconds: 40
   ```

3. DNA sequence searching is a common task in computational biology. Create a function (**`listSequencePositions`**) to find the locations of a given string within a larger string. Your function will take two parameters, the sequence to be found and

the larger sequence to be searched. Your function will return all the location(s) where a sequence match was found in a list.

If your function is called with "CCG" and it is searching the human DNA from assignment #4, your output should correspond to the following:

```
11 61 98 165 179
```

4. The story generator – the Mad-libs game asks for parts of speech, such a noun, adjective, or adverb, and those words are plugged into a template to generate a sometimes-funny story.

   For this problem, write a function that plays a game of Mad-libs. Your function (**`MadLib`**) will be given an integer parameter to select the story from a list of story templates and will return the MadLib string with words from the user substituted for the placeholders. You program will store the templates as strings and ask the user for appropriate parts of speech to fill in the template based on the given parameter. Your program needs to include the following templates.

   1. "Be kind to your <noun>-footed <plural noun>, or a duck may be somebody's <noun>."
   2. "It was the <adjective1> of <noun1>, it was the <adjective2> of <noun2>."
   3. <plural noun>? I don't have to show you any <adjective> <plural noun>!
   4. My <relative> always said <noun> was like a box of <noun>. You never know what you're gonna get.
   5. One <time of day>, I <verb> a <noun> in my pajamas. How he got in my pajamas, I don't know.

   If for example, the function is called as MadLib(2), your function should ask for entries for each of the placeholder words in the template, such as:

```
Enter adjective1:
Enter noun1:
Enter adjective2:
Enter noun2:
```

   Each template will require multiple words of differing types to be inserted into the string. Substitute the user words for the placeholder and return the completed string.

Challenge Problem I

Modify your DNA search to read the search DNA from a given filename. Return the list of positions as a Python list instead of printing those locations.

Challenge Problem II

Add testing code to randomly play a MadLibs game each time the user agrees to play. When the program starts, you should first ask the user if they want to play a game. Your question should look like:

"Do you want to play a game? (y) or (n)"

If the user types "n" when asked if they want to play, your program should print "good bye" and exit. Otherwise, randomly select a game (number between 1 and 5) to be played and call the ***MadLib*** function with that value. Print out the resulting sentence to the console and ask again if the user wants to play.

Challenge Problem III

Read the templates from a file. Each line in the file would store a single template. Read all the templates into a list of strings. Change your loop to randomly select from the list of strings. There are several on-line Mad-Libs games that you can use for creating your own templates.