Python Population Estimate Project

Assignment Overview:

This assignment focuses on the design, implementation and testing of a Python program to solve the problem described below.

DUE: Completed program no later than 5:30 PM on Wednesday, April 10, 2019.

Assignment Specifications

1. This programming project will estimate the population of the United States based on some information provided by the US census. (Newer data can be found at https://www.census.gov/popclock/)

   For this project, use the following data:

   - A number that represents the current population. Use the following number for the starting point for this project at the value: 307357870
   - 3 rates:
     - every 7 seconds, a birth
     - every 13 seconds, a death
     - every 35 seconds, a new immigrant

   Predicting into the future using only these numbers is not very accurate since the rates are likely to change, but it does give you a general idea for short predictions (only a few years into the future). Longer predictions (100's of years) are likely to be more inaccurate.

2. Your program will prompt the user for the number of **years** into the future we are predicting. Your program then prints out the predicted population size as an **integer**. You must use the above-indicated values: the starting point and the rates. The input value must represent years.

   Use the following information about Time Conversion:

   - 365 days in a year (not quite true, but let's use it anyway)
   - 24 hours in a day
   - 60 minutes in an hour
   - 60 seconds in a minute

3. Calculate and Output. Use the sample data in the graphic below to test your program and output as described.

   - How many years
   - New population

4. Handle Errors

   - If trying to divide by zero?
   - If entered a letter instead of a number at the prompt?

5. Repeat #2 until the user indicates an action to STOP.

Suggested Procedure

- Break the problem down into smaller steps and start with a simple version of the program.
- You may want to solve the problem using pencil and paper first. You cannot write a program until you have figured out how to solve the problem.
- Cycle through the steps to incrementally develop your program.
- Edit your program to add new capabilities.
- Run the program and fix any errors.