Programming Assignment #4
Summer 2017
Math for Fifth Grade:

Overview

In this program you are going to build a python program that uses the graphics library (created by Dr. Zelle) you learned in the class to help the kids in the fourth grade to learn the basic mathematical operations (addition, subtraction, multiplication and division). This program will display several problems -15 problems- where each one uses the random library to generate the numbers and the problem operation. The program then allows the user-kid- to enter his/her answer. After that your program should compare the user's answer with the right one and then give an instance and accumulative feedback for the kid on his current answer and his overall performance so far. After finishing all the problems, your program should display some statistics about the kid's performance.

In this program, you are expected to use the principles of random numbers, iterations (loops), Selection (if), functions and the graphics library. You should define and use several functions to write this program. The program uses the random number library to the operands (the two numbers) of the operations and also to choose the mathematical operations itself (addition, subtraction, division, and multiplication). The two generated numbers for the problem should have values between 0 and 30 and you have to make sure that the problem result is a whole integer number between 0 to 100 (Be careful that in the division operation the result should not include a fraction), in case it includes then you have to change the problem and your program should generate another problem and make sure that the new problem follow the previous rule too.

Program Design:

Read the following pseudocode that describes how the program is going to run.

- Your first step to do is to create a python file name it "program3.py".
- Write the **program prolog** as **comments** in the first few lines of your program. (Use plain English, not python code and use # at the beginning of the program to write the comments.) The program prolog includes your name, date and time you finished working on the program and a description of your program.

- Inside your program write the following design as comments. After the # symbol, give every step of your design a number so you can refer to it later on.
- This is a suggested design for the program.

  1. Inside the main function, create a graphics window 600*600.

  2. Call a function DisplayWindow and pass it to one argument which is the name of the window you created in step 1 that will make the window background to be blue.

  3. Then inside the main use for loop to ask the user for 15 questions.

  4. Inside the for loop do the following:

     A. Call a function GenerateNumber that takes no parameters and returns one value which is a random number between 0 and 30. Store the returned value inside a variable called number1.

     B. Call a function GenerateNumber again and store the returned value inside a variable called number2.

     C. Call a function GenerateOperation that takes no parameters and returns one value which is a random number between 1 and 4 (1 and 4 are included). Store the returned value inside a variable called operationType.

     D. Call a function VerifyOperation that takes three parameters and returns two values. The passed arguments are number1, number2 and operationType. And the function returns two values (The first one is a Boolean value –either True or False- and the second one is numerical value – either -1 or the actual result of the operation–. Store the returned values into two variables under the name ValidOrNot and ActualResult.

     E. If the value returned from 4.D is false, then you have to write a while loop and repeat steps 4.A 4.B and 4.C

     F. Otherwise, you have to call function DisplayProblem that takes six parameters (The passed arguments are the window name, number1, number2 and operationType, ActualResult, and the current problem number) and it returns a Boolean value. Used the returned value to increase the counter of the value of a variable called CorrectAnswer by one. CorrectAnswer should be given the value zero before the for loop, CorrectAnswer is a counter that calculates the number of correct answers

5. After you finish showing and calculating the 15 problems call a function DisplayStatistic and pass two arguments. The first one is the window name and the second one is the CorrectAnswer variable.

6. The program then waits for a click before terminating the game.

Functions:

Function ***DisplayWindow*** takes one parameter which is the name of the window coming from the main function. This function set the window background color to blue.

Function ***GenerateNumber*** takes no parameters and returns one value which is a random number between 0 and 30.

Function ***GenerateOperation*** takes no parameters and returns one value which is a random number between 1 and 4.

Function ***VerifyOperation*** takes three parameters and it returns two values. The passed arguments are number1, number2 and operationType. And the function returns two values (The first one is a Boolean value –either True or False- and the second one is numerical value under the name ActualResult – either -1 or the actual result value of the operation–). The function calculates the mathematical operation between the two numbers based on the passed operationType. If the result is less than 0 or more than 100 then returns backs two values, False and -1, otherwise the function returns back two values; True and The actual result value.

Function ***DisplayProblem*** takes six parameters (The window name, number1, number2 and operationType, ActualResult, and the current problem number) and it returns a Boolean value. The function displays three text boxes that show the two numbers and the operation between them. Also it shows an Entry for the user to input his/her answer. The data input by the user inside the entry should be store under the name UserAnswer. After the user enter his/her value and click a mouse. Compare the userAnswer with the ActualResult and store the results under name ValidateAnswer. Call function DisplayProgressBar and pass to it (the window name, the current problem, and ValidateAnswer) after that let the function DisplayProblem returns the value of ValidateAnswer.

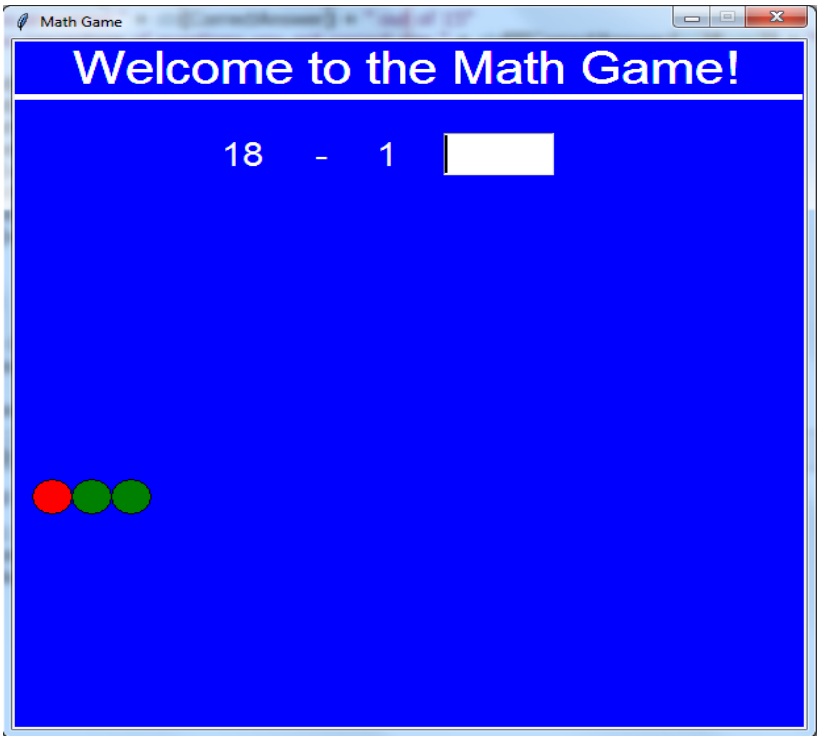
Function ***DisplayProgressBar*** takes three parameters (The windows name, the current problem number, and the ValidateAnswer). The function does not return anything. The function first shows a text book that displays “Right Answer” or “Wrong

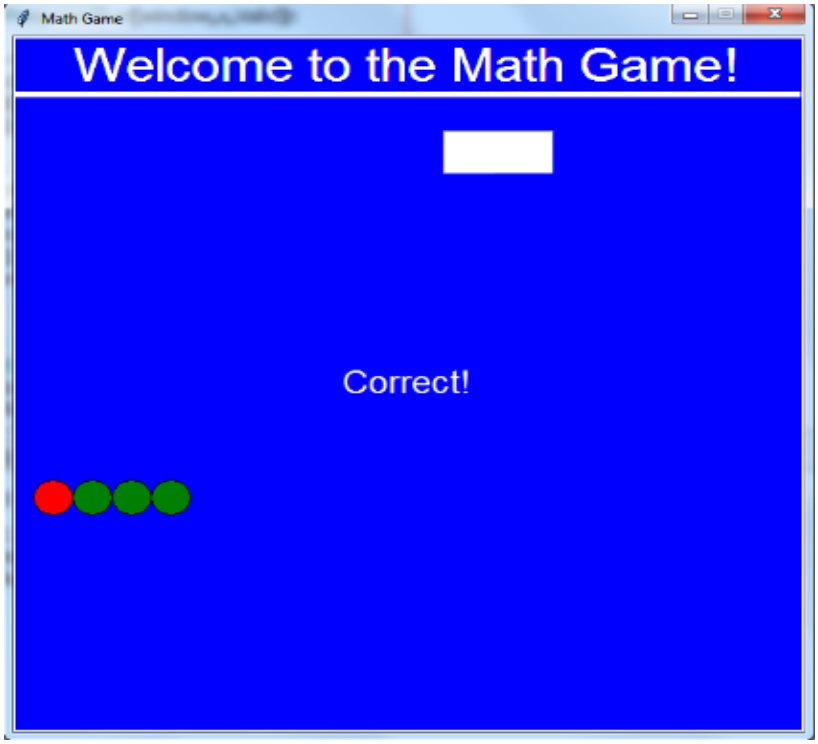
Answer" based on the value of ValidateAnswer and then it displays ProgressBar at the bottom of the window – you can express that as circles or squares, be creative-. The shape you choose should be filled with the color green if the ValidateAnswer is True. Otherwise fill the chosen shape by red color. Wait for 1 second then undrawn the text box only.

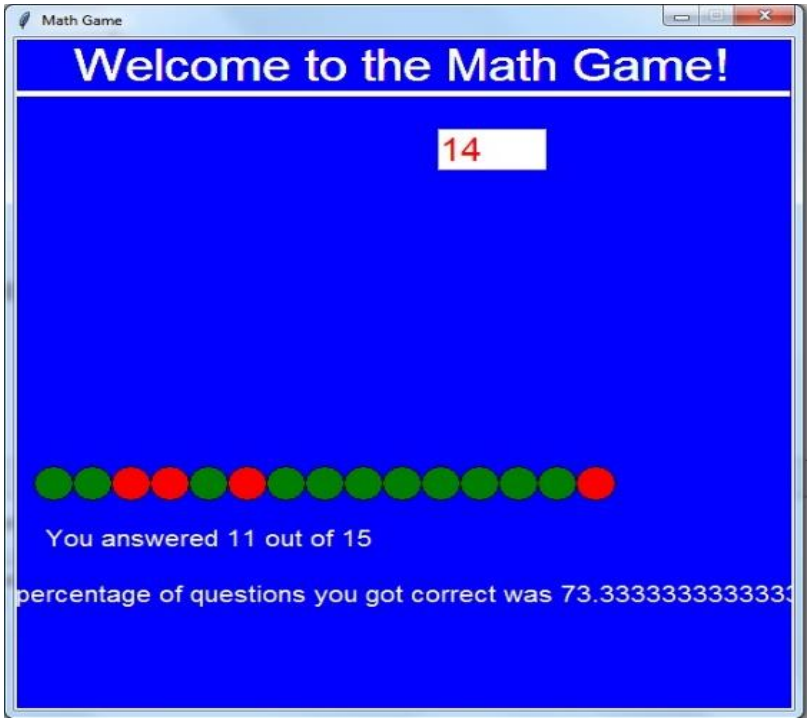
Function ***DisplayStatistic*** takes two parameters (The window name and the second one is the CorrectAnswer). The function display a summary of the correct problem, the wrong problems, and the percentage of the correct answers comparing to the total number of problems.

<u>See the following snapshots from different run for the program:</u>

\







DUE DATE:

The due date for submitting your program assignment code is Wednesday 7/26 midnight. Submit your work on CANVAS.