IS2053-Fall 2022 – Lab02

Complete and upload the following program to CodeGrade through BlackBoard as a Python file named **lab2.py**.

Write a program that creates a text file (randnums.txt) that stores a user-defined amount of random numbers. The program would then open randnums.txt and read its contents. As each number is read from the file, the program should check to see if the number is divisible by 2, divisible by 3, divisible by 5, and divisible by 10. The results should be printed line-by-line, as shown in the example below.

Program Objectives:

- Import random module
- Comments – lots of them!
- Local variables (lowercase)
- Global variables (UPPERCASE)
- Properly reading keyboard input
- Calculations
- Comparing strings
- Logical operators
- Properly formatted output using f-strings
- The if-else decision structure
- The while Loop: A Condition-Controlled Loop
- The for Loop: A Count-Controlled Loop
- Calculating a Running Total
- Input Validation
- Defining and Calling a Void Function
- Defining and Calling a value-returning Function
- Passing Arguments to Functions
- Using Loops to Process Files
- Processing Records
- Exceptions (try/except)

Your program will:

1. Import the random library.
2. Use the global constants RAND_LO = 0 and RAND_HI = 999 to hold the lower and upper boundaries for the random numbers to be generated.
3. Prompt the user to enter a number between 1 and 100.
4. Use an input validation while loop to ensure the number entered is between 1 and 100. The prompt should continue until an acceptable number is entered.
5. Open (create) randnums.txt for writing.
6. Use a for loop to write the user-defined amount of random numbers to the output file.
    a. The number of lines written is determined by the user's entry in step 3. For example, if the user enters 15, there should be 15 lines of random numbers in your file.
    b. Use the global constants from step 2 for the lower and upper boundaries of the random numbers to be generated.

7. Close the output file.
8. Open randnums.txt for reading.
9. Process the input file with a while loop. (A priming read will be necessary.)
10. Keep a running total of the numbers read from the file.
11. Keep a running count of the number of lines read.
12. Print the values of each line by using the print_line function (see below)
13. The while loop should continue until all lines are read from the file.
14. Print a header and footer for the table that is generated. (see example below)
15. Print the value of the running total and the average below the footer. (see example below)
16. Close the input file.
17. Utilize a Try/Except block with exceptions for IOError and ValueError.
18. Define a function named print_line with the parameters of num_in and num_ln.
    a. The function does not return a value.
    b. The function prints one line each time it is called with the following specifications:
        i. The line number from the num_ln variable.
        ii. The number from the num_in variable.
        iii. True/False based on the output of div_by_2 function.
        iv. True/False based on the output of div_by_3 function.
        v. True/False based on the output of div_by_5 function.
        vi. True/False based on the output of div_by_10 function.
        (See function descriptions below – pass the num_in argument to these functions.)
19. Define a function named div_by_2 that accepts an integer and returns the Boolean value True if the integer is divisible by 2, otherwise False.
20. Define a function named div_by_3 that accepts an integer and returns the Boolean value True if the integer is divisible by 3, otherwise False.
21. Define a function named div_by_5 that accepts an integer and returns the Boolean value True if the integer is divisible by 5, otherwise False.
22. Define a function named div_by_10 that accepts an integer and returns the Boolean value True if the integer is divisible by 10, otherwise False.
23. Hint: To determine whether a number is divisible by another number, refer to page 53 in chapter two and explore the remainder operation. If one number is divided by another number and has a remainder of zero, then it is true that the first number is divisible by the second number.
24. Another hint: get one of the div_by_* functions working, copy/paste them and change the numbers.

Here is an example of how your program's output should appear:

```
How many random integers should be written to the file?
Please enter a number between 1 and 100: 101
Please enter a number between 1 and 100: 0
Please enter a number between 1 and 100: -6
Please enter a number between 1 and 100: 900
Please enter a number between 1 and 100: 15

-------------------------------------------------
|Line | Num | Div/2  | Div/3  | Div/5  | Div/10 |
-------------------------------------------------
|  1  | 762 | True   | True   | False  | False  |
|  2  | 110 | True   | False  | True   | True   |
|  3  | 338 | True   | False  | False  | False  |
|  4  | 160 | True   | False  | True   | True   |
|  5  | 361 | False  | False  | False  | False  |
|  6  | 716 | True   | False  | False  | False  |
|  7  | 311 | False  | False  | False  | False  |
|  8  | 832 | True   | False  | False  | False  |
|  9  | 854 | True   | False  | False  | False  |
| 10  | 746 | True   | False  | False  | False  |
| 11  |   3 | False  | True   | False  | False  |
| 12  | 328 | True   | False  | False  | False  |
| 13  | 688 | True   | False  | False  | False  |
| 14  | 622 | True   | False  | False  | False  |
| 15  | 891 | False  | True   | False  | False  |
-------------------------------------------------
Total: 7,722  Average: 514.80
```