SE 5930
Machine Problem 3 (MP3)
Addition Practice

20 points

We have just learned how to use **while-loops**, and define our own **functions**. Write a Python script that implements a **while-loop** and defines 2 **functions** in order to produce an Addition Practice session for the user. Your program output must look exactly like the following output (except the numbers in the problems will be randomly generated)…

```
ADDITION PRACTICE PROBLEMS

Add the numbers together and enter your answer.
I'll tell you if you are right or wrong.

Problem 1: 58 + 22 = 80
You are right!

So far... 1 right, 0 wrong.

Do you want to do another problem? ('Y' or 'N') y

Problem 2: 56 + 35 = 91
You are right!

So far... 2 right, 0 wrong.

Do you want to do another problem? ('Y' or 'N') yes

Problem 3: 64 + 88 = 123
Sorry, that is not correct.  The correct answer is 152.

So far... 2 right, 1 wrong.

Do you want to do another problem? ('Y' or 'N') Y

Problem 4: 29 + 72 = 100
Sorry, that is not correct.  The correct answer is 101.

So far... 2 right, 2 wrong.

Do you want to do another problem? ('Y' or 'N') Yes

Problem 5: 19 + 84 = 103
You are right!

So far... 3 right, 2 wrong.

Do you want to do another problem? ('Y' or 'N') YES

Problem 6: 66 + 33 = 100
Sorry, that is not correct.  The correct answer is 99.

So far... 3 right, 3 wrong.

Do you want to do another problem? ('Y' or 'N') n
```

- Your program solution must include 2 **functions**. The details of the **functions** are in the documentation. This documentation must be included in your code…

```
#
# Machine Problem 3
#
# Mike Liljegren
#
# Description: This script is addition practice problems.  The user is given 2 integers
#              to add together.  The user gives an answer and is told if the answer is
#              correct, or not.  The user is then asked if they want to do another problem.
#              The number of correct and the number of incorrect answers are recorded
#              after each problem.
#

from random import *

def additionProblem(problemNum):

    #
    # Prompts the user with an addition problem with 2 random int values between
    # 1 and 100.  Collects the answer from the user, and generates the correct
    # answer.
    #
    # problemNum    An int value.
    #
    # Returns a tuple with 2 elements.  The 1st element is the user's
    # answer(int) and the 2nd element is the correct answer (int).
    #
```

< the body of your function `additionProblem()` **goes here >**

```
def gradeAnswer(answers, answerTally):

    #
    # Checks to see if the user's answer and the correct answer are the same.
    # Prints the appropriate message to the user, along with a running total
    # of how many right and wrong answers the user has accumulated in this
    # session.
    #
    # answers       A tuple with 2 elements.  The 1st element is the user's
    #               answer (int) and the 2nd element is the correct answer (int).
    # answerTally   a list with 2 elements.  The 1st element is a running total
    #               of how many right answers so far in this session (int),
    #               and the 2nd element is a running total of how many wrong
    #               answers so far in this session (int).  The list is modified
    #               in the function.
    #
    # There is no return value.
    #
```

< the body of your function `gradeAnswer()` **goes here >**

```
### Main Program ###
```

< the body of your main program goes here >

- Your solution must include a **while-loop** in the main program. The main program will call both **functions**.

- The output that you hand in must show examples that your program will accept ‘y’, ‘Y’, ‘yes’, ‘Yes’, and ‘YES’ to continue doing more addition problems.

- Your output must show both correct and incorrect answers being handled and counted.

For full credit…

- Hand in a hard copy of your script and the output that your script generates.
- Hand in your solution on-time.
- Document your solution appropriately.
- Write clear, well-organized code.
- Use variable names that make sense for the data that they contain.
- The output that your script generates must be exactly like the desired output (of course the numbers will be different because they will be randomly generated).