

day31 review 2 for Quiz #2

Due: Wednesday 11/1/23

Today is more practice for our quiz that is next class. Doing this assignment well will help you with the quiz. Create a Python replit assignment named day31 review. Please print the part number for each part before you work. For example, "Part 1:" then on the next line do part 1, etc.

1. Print "Part 1:" then do this part. I put some test scores into a list variable as shown. Write code that **uses a for loop** to go through the list, add up all the values, and then report the average as well as the largest value in the list. Your code must work for any list, not just this exact list (so instead of hard-coding that there are 9 items in the list you would use `len(scores)` to find out how many items are in the list. Use this actual data (copy this line from the PDF). To find the average of some numbers, add up all of the numbers and divide by how many numbers there are. Use code to find the maximum value, don't just print 98 because you can see that it is the largest value.

```
scores = [81,45,98,53,64,84,77,60,93]
```

The correct answer for the average of these numbers is `72.77777777777777`. The idea of this part is that if I changed the list of numbers, your code would still find the average and largest value accurately.

2. Use a for loop to print 15 random numbers (each between 10 and 13 inclusive) all on the same line. Then print a blank line.
3. Write a for loop to print the numbers 98 through 140 in 4 columns separated by tabs.
4. Write a function that accepts a piece of text and then prints it in a box as shown here. Your code works with any length text. Add tests to print two different phrases that have different lengths. Do not ask the user to enter anything. The horizontal line is all the letters of the text with a space between each letter.

```
*****  
* H a y s *  
* a *  
* y *  
* s *  
*****
```

5. Write code to create a list with two of your classes at Analy in it. Then use the `.append` method to add another class to the list. Sort the list. Then print the list using a for loop, one class per line. Do not use a `break` command. Do not ask the user to enter anything.
6. Write a for loop that prints the numbers 13 to 21, one number on each line.
7. Write code that uses a `try/except` structure with an input command to request an integer value between 7 and 14 inclusive. Include one error message to the user on `ValueError`, another if the number is too small, and a third one if the number is too large. Once the user enters a valid number, print "Thanks" that many times using this line: `print("Thanks"*num)`

(continued on next page)

8. Write a function `doit(name)` which accepts a name and prints this:

```
Hays  
.Hays  
..Hays  
...Hays
```

The function takes whatever is sent to it and prints it, then prints one period and prints the text again, then two periods, then the text, again, until it has printed the text as many times as the text has characters. You will end up with one fewer period than the length of the text (the last line has three periods, and "Hays" is four letters long.) After you write the function, add two tests with phrases of different lengths.