Practice Exam 2

Python

Answer the questions in the spaces provided. **Please note** that there are no intentional errors in the code provided except in questions asking you to correct said code.
Your written code does not have to be 100% syntactically correct.

Name: ________________________________

Page	Points	Score
4	20	
7	10	
8	15	
9	10	
10	10	
12	20	
13	15	
Total:	100	

Useful notes:

- The practice exam is longer and harder than the actual exam. It was originally created for a 2 hour exam on paper, which can be done quicker than on a computer.
- I've stuffed this exam full of extra practice questions. If you're timing yourself, ignore them until later.
- The points roughly correspond to the difficulty of the questions.
- If it's more than 10 points, it's very important.
- The amount of space given for a problem **does not** correspond to the difficultly of the problem.
- You are allowed to clarify any answer you give.
- You are allowed to ask for clarification.
- Things are never as complicated as they appear, especially the math.
- Never leave a question blank, even if you don't know the answer. We can't give partial credit to blanks.

Don't Panic

__________ 0 points

1 Short Answer

These kind of questions won't be on the exam, but are good review.

1. (0 points) The string value "Howl's Moving Castle" is a valid string. Why isn't it a problem that the single quote character in the word Howl's isn't escaped?

2. (0 points) What is the main difference between a dictionary and a list?

3. (0 points) If l is a list, what does the expression `l[::-1]` do?

4. (0 points) How can use slice notation to get the last 3 characters of a string?

5. (0 points) How can you simulate a coin flip in python?

6. (0 points) I have a coin that is slightly weighted in favor of heads; it will land on heads 55% of the time. How would you simulate this in python using the random module?

2 File Reading

7. (20 points) **From A Prior Exam** Suppose we had a file called `temperature.csv`, which contains the daily high and low temperatures recorded in Philadelphia. The file is composed of three fields per a line, containing the date, the high temperature, and the low temperature. Each field is separated commas. For example, the contents of the file might look like this:

```
11/11/19,  64,  40
11/12/19,  49,  21
11/13/19,  33,  22
...        ...  ...
```

Write a program that calculates or prints the following:

5pts The day with the hottest high temperature.

5pts The day with the coldest low temperature.

10pts The average high temperature of the entire file.

Your program on this page:

______________ 0 points

8. (0 points) **This question is extra practice.**

   Suppose you have a file named `numbers.csv` which contains a bunch of integers, five per line of text, separated by commas. Write code below that will open the file, read the numbers from it, and print the sum of all the **even numbers** in the file. For example, the file might look like this:

```
353,213,5,12399421,1
-4,3243,2323456,32186,4234
123,1,2,3,4
0,8,6,-3,2
```

3 Method Writing

9. (5 points) `unOrUn`: If the given String begins with "un", return a String with without the "un" in front. Otherwise, return the String with "un" added to the front of it. You may assume the String is at least 3 characters long.

```
# unOrUn("untied")    -> "tied"
# unOrUn("unable")    -> "able"
# unOrUn("necessary") -> "unnecessary"
def unOrUn(word):
```

10. (5 points) `maxMinDiff`: Given an list of integers, return the difference between the maximum element and the minimum element. You may assume the list will have 2 or more elements in it.

```
# [1,2,3,4,5]  -> 4
# [15,31,21,17,28] -> 16
# [-1,-100,12,2,100] -> 200
def maxMinDiff(numbers):
```

__________ 10 points

11. (0 points) **Extra Practice** `swapEnds`: Given a list of numbers, modify the list such that the first and last elements are swapped. You may assume will have 2 or more elements in it.

```
# [1,2,3,4,5]  -> [5,2,3,4,1]
# [15,31,21,17,28] -> [28,31,21,17,15]
# [-1,-100,12,2,100] -> [100,-100,12,2,-1]
def swapEnds(numbers):
```

12. (5 points) `firstHalf`: Given a list of things, return the first half of the list.

```
# [1,2,3,4,5,6]  -> [1,2,3]
# [15,31,21,17,28] -> [15,31]
# ["a","b","c","d","e","f","g","h"] -> ["a","b","c","d"]
def firstHalf(theList):
```

13. (10 points) Write a method called `hasWildcat`: Given an input String `word`, return true if `word` contains the String "cat" in it, but the middle 'a' can be any `char`.

```
# hasWildcat("kitty")  -> false
# hasWildcat("tomcat") -> true
# hasWildcat("c4tn1P") -> true
def hasWildcat(word): #  A-- would not buy again
```

14. (0 points) **Extra Practice** We'll say that a value is "everywhere" in an array if for every pair of adjacent elements in the array, at least one of the pair is that value. Return true if the given value is everywhere in the array.

```
# source: http://codingbat.com/prob/p110222
# isEverywhere([1, 2, 1, 3], 1) -> true
# isEverywhere([1, 2, 1, 3], 2) -> false
# isEverywhere([1, 2, 1, 3, 4], 1) -> false
def isEverywhere(nums, val):
```

15. (10 points) Given a string, return the sum of the digits 0-9 that appear in the string, ignoring all other characters. Return 0 if there are no digits in the string. (Note: `s.isdigit()` tests if a the string `s` is one of the characters '`0', '1', ... '9'.`)

```
# http://codingbat.com/prob/p197890
# sumStringDigits("aa1bc2d3") -> 6
# sumStringDigits("aa11b33") -> 8
# sumStringDigits("Chocolate") -> 0
def sumStringDigits(word):
```

16. (10 points) Given an integer, return the sum of the digits 0-9 of that integer.

```
# sumDigits(1234) -> 10
# sumDigits(1000) -> 1
# sumDigits(-581) -> 14
def sumDigits(num):
```

17. (0 points) **Extra Practice** Given a list of positive numbers, return a new list of length “count” containing the first even numbers from the original array. The original list will contain at least “count” even numbers.

```
#http://codingbat.com/prob/p134174
#copyEvens([3, 2, 4, 5, 8], 2) -> [2, 4]
#copyEvens([3, 2, 4, 5, 8], 3) -> [2, 4, 8]
#copyEvens([6, 1, 2, 4, 5, 8], 3) -> [6, 2, 4]
def copyEvens(nums, count):
```

From Prior Exam In the board game Monopoly, players move around the board by rolling a pair of six-sided dice. Rolling “doubles,” such as two 3’s or two 5’s, is especially fortunate, as this allows the player to roll again, essentially getting the player an extra turn. If the player rolls another set of doubles[1], this allows the player yet *another* extra turn. However, rolling doubles a third time will immediately land the player in “jail,” ending their turn and trapping the player until they can roll a double.

Write a program that will simulate 100000 Monopoly turns (extra rolls due to doubles count as the turn), where a turn is a single roll of the dice or more if the user rolls a double. Your program should print out:

- The percentage of rolls without a double.
- The percentage of rolls with 1 double.
- The percentage of rolls with 2 doubles.
- The percentage of rolls with 3 doubles.

3.1 Grading

Points will be given as follows

10pts Simulating 100000 dice rolls.

5pts Handling doubles.

3pts Handling two and three doubles in a row.

2pts Presenting the data in percentages.

[1]It can be the same pair as the first time

18. (20 points) Put your program here:

______________ 20 points

19. (15 points) Given a string, write a function which returns the most common letter in the string. Use a dictionary to do this.

Extra Practice The outcome of any fair coin toss has an equal chance of landing on heads or tails. If I were to toss a coin three times in a row, there are a total of 8 equally probable outcomes.

```
HHH
HHT
HTH
HTT
THH
THT
TTH
TTT
```

where H is heads and T is tails. Thus, `HHT` represents two heads thrown in a row, followed by a tails.

Suppose Professor Rosen challenges you to coin flipping game for $100. You will choose a sequence that occurs after three coin flips, such as `HHT`, and Professor Rosen will choose one, such as `THH`. One of you will keep flipping a coin until one of our three flip sequences occurs, and whoever's sequence shows up first wins. So, for example, if we flip the coin until one of the chosen example sequences occurs, a few games might be:

```
HTHTHTHH -> Rosen wins
HHT -> You win
TTHTHH -> Rosen win
HHHHHT -> You win
```

This seems like a fair bet; after all, all eight outcomes of flipping a coin three times are equally likely. Then you remember that Professor Rosen mentioned he was sorted into Slytherin and is a bit of a trickster; he probably wouldn't be challenging you unless he was sure he was going to win.

So let's write a program **on the next page** to simulate this game and see if your professor is trying to scam you.[2]

[2]Once you finish the practice exam, you should learn more about the game by watching Numberphile's video on Penney's Game, found here: `https://www.youtube.com/watch?v=Sa9jLWKrX0c`

20. (0 points) Write a program that simulates the game on the previous page with one player trying to get `HHT` and the other trying to get `THH`. [3] Have your program simulate the game 100000 times and print out the percentage of times the two players win.

[3] Other games to simulate if you want to see more is `TTT` vs `HTT` and `HHT` vs `HTH`