CS 2316 Individual Homework 2 – Conditionals & Loops
Due: Wednesday, January 22nd, before 11:55pm
Out of 100 points

File to submit: HW2.py

Students may only collaborate with fellow students currently taking CS 2316, the TA's, and the lecturer. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc.

For Help:

- TA Helpdesk – Schedule posted on class website.
- Email TA’s or use Piazza

Notes:

- **Don’t forget to include the required comments and collaboration statement (as outlined on the course syllabus).**
- ***Do not wait until the last minute*** *to do this assignment in case you run into problems*
- **Read the entire specifications document before starting this assignment.**

Simple Functions
You will write a few python functions for practice with the language. In your HW2.py file, include a comment at the top with your name, section, GTId/Email, and your collaboration statement. Also, include each of the following functions below. For purpose of this homework, you may assume that all inputs will be valid.

1. **countVowels**
2. **gradeReplacement**
3. **cellphoneBill**
4. **strElongation**
5. **numHourglass**
6. **gpaCalculator**
7. **nextRow**
8. **clockTurtle**

1. countVowels(10pts)

Description:
Write a function that takes in one parameter: a string. Your function will analyze the string and print out the number of occurrences of each vowel respectively (a, e, i, o, and u). Note with caution that your function should be able to recognize both uppercase and lowercase vowels! The format of the information to be printed can be found under “Test Cases”.

Parameters:
aString (String): A string

Return Value:
None

Test Cases:

1. countVowels(“so it goes”) prints “a: 0, e: 1, i: 1, o: 2, u: 0”
2. countVowels(“Are you enjoying this so far?”) prints “a: 2, e: 2, i: 2, o: 3, u: 1”
3. countVowels(“CS is sOoOoOo fun!”) prints “a: 0, e: 0, i: 1, o: 6, u: 1”

2. gradeReplacement (10pts)

Description:
Write a function that takes in a list of exam grades and returns the average after performing a grade replacement policy. The grade replacement policy takes the lowest grade from the list and replaces it with the second lowest grade in the list. There will be at least 2 numbers in the list, but there is no upper bound for the length of the list.

Parameters:
gradeList (List): A list of exam grades as integers

Return:
(Float) The average of all the exam grades after replacement

Test Cases:

1. gradeReplacement([100,90,80,70]) returns 87.5
2. gradeReplacement([100,100,100,100]) returns 100.0
3. gradeReplacement([90, 80, 80, 90, 85]) returns 85.0
4. gradeReplacement([30, 80, 44, 90, 85]) returns 68.6

3. cellphoneBill (10pts)

Description:
Write a function that takes in a list. The elements in the list are integers representing the number of minutes each person/phone on a cellphone plan used last month. Calculate the amount owed based on the following criteria: the first 250 minutes used on each phone has a flat rate of $25 (if the amount of minutes used is less than 250, the charge is still$25). The next 250 minutes (251-500) are an additional $0.02 per minute charge. From minute 501 onward, each minute is an additional $0.05 charge. However, if a plan has 2 people/phones on it, the sum of their charges is discounted by 5%. Similarly, if a plan has 3 people/phones on it, the sum of their charges is discounted by 10%. Likewise, plans with 4 or more people receive a discount of 15% of the sum of their charges. Return your floating point answer with no more than 2 digits after the decimal point (it is a dollars & cents answer).

Parameters:
minutesList (List): A list of minutes used per person/phone last month

Return Value:
(Float): The total amount needed to be paid

Test Cases:
1. cellphoneBill([25,900,251]) returns 90.02
2. cellphoneBill([1]) returns 25.0
3. cellphoneBill([555,1032,400,1429]) returns 164.73
4. cellphoneBill([250,500]) returns 52.25
5. cellphoneBill([249,1301,1350,2316,4400])]) returns 436.35

4. **strElongation(10pts)**

Description:
Write a function that takes in a string and prints out an elongated version of it. Each item in the string should be multiplied by its position in the string. See example test cases below for clarification.

Parameter:
aStr (String): The string to be elongated

Return Values:
None

Test Cases:

1. strElongation(“Jake”) prints “Jaakkkeeee”
2. strElongation(“I <3 2316”) prints
   “I <<<3333   222222333333311111111666666666”
3. strElongation(“telescopic”) prints
   “teellleeeesssssccccccoooooooppppppppiiiiiiiiicccccccccc”

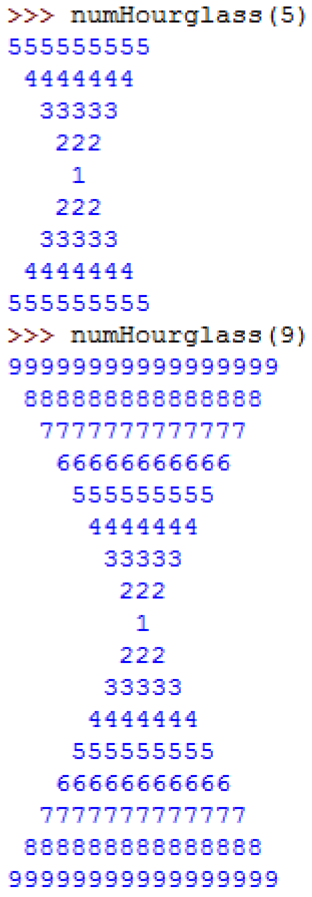
5. numHourglass(15pts)

Description:
Write a function that will take in an integer X and return an hourglass with a maximum width of 2X-1. Each row will be made of integers representing said row, see the output examples below for clarification.

Parameters:
aNum (Int): An integer representing the maximum width of the hourglass

Return Value:
None

Test Cases:
Notice that you have three 2’s, seven 4’s, nine 5’s, etc.



```
>>> numHourglass(5)
555555555
 4444444
  33333
   222
    1
   222
  33333
 4444444
555555555
>>> numHourglass(9)
99999999999999999
 888888888888888
  7777777777777
   66666666666
    555555555
     4444444
      33333
       222
        1
       222
      33333
     4444444
    555555555
   66666666666
  7777777777777
 888888888888888
99999999999999999
```

5. **gpaCalculator (20pts)**

Description:
Write a function that takes in 2 lists of equal length. The first list will contain numbers from 0-100 representing final grades in different classes. The second list will represent the number of credit hours each class grants upon completion. The indexes in both lists coincide. For example, if the first element in the first list is a 97 and the first element in the second list is 3, this data represents a 3 hour class that one received a grade of 97 in. Using this data you will need to calculate the number of quality points received from each class. Quality points are calculated as the grade point achieved in the class times the number of credit hours of that class. Grade point are determined as follows: 4 for [90,100], 3 for [80,90), 2 for [70,80), 1 for [60,70) and 0 for [0,60). Once the quality points have been determined, the grade point average is calculated as the sum of quality points over the sum of the credit hours.
Your function should return the calculated GPA.
Parameters:
gradesList (List): This list will hold finals grades (out of 100)
hoursList (List): This list will hold the amount of course hours granted
Return Value:
(Float): A floating point value representing the grade point average.

Test Cases:

1. gpaCalculator([87,81,91,99,93],[3,3,4,3,3]) returns 3.625
2. gpaCalculator([74,90,88,40],[3,4,4,3]) returns 2.4286
3. gpaCalculator([100,30,50,75,90],[3,3,1,4,2]) returns 2.153846
4. gpaCalculator([59,82,74,79,66],[4,3,3,1,3]) returns 1.42857

7. nextRow (10pts)

Description:
This is the beginning of pascals triangle:

```
         1
       1 1
     1  2 1
  1  3  3  1
1  4  6  4  1
```

You can calculate any row of Pascal's triangle after the first two from the previous row. Create a new row that starts with a 1, filling in the inner values such that each number is the sum of the two values to the upper left and upper right above t in the previous row, and then adding a 1 to the end.

For example, the 3rd row of Pascal's triangle is [1,3,3,1]. So the 4th row would be calculated as [1, 1+3, 3+3, 3+1,1] = [1,4,6,4,1].

Write a function called **nextRow** that takes in one parameter, a list of numbers representing a row in a Pascal's triangle (of at least 2 numbers) and return a list representing the next row.

Parameters:
aRow (List): A row of integer numbers.

Return Values:
(List): A list of numbers representing the NEXT row of the triangle.

Examples:
nextRow([1,1]) returns [1,2,1]
nextRow([1,2,1]) returns [1,3,3,1]

8. clockTurtle (15pts)

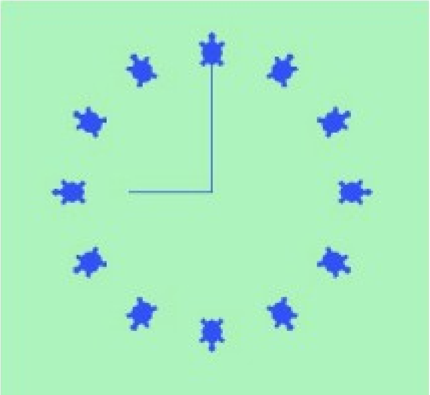
Description:
Write a function that uses the turtle module to draw a clock with a given clickHour, as the short hand of clock, and aNum, as the radius. You may assume that the long hand of the clock will stay at 12 at all times. You do need to draw the clock layout using the turtle module. At each hour position (12, 1, 2, 3, 4, 5, etc.), make your turtle leave a stamp of itself. (You can change the turtle shape if you want).

Parameters:
clockHour (Integer): an integer between 1 and 12 representing the short hand of the clock
aNum (Integer): radius of the clock

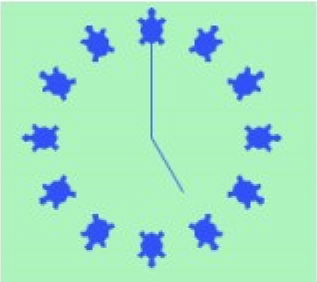
Return Values:
None

Examples:

clockTurtle(9,100)



clockTurtle(5,70)



Grading Rubric

countVowels (10pts)

- Loops through input string	2pts
- Identifies lowercase vowels	2pts
- Identifies uppercase vowels	2pts
- Keeps track of the number of occurrences of each vowel	2pts
- Prints correct information in the correct format	2pts

gradeReplacement(10pts)

- Correctly identifies the lowest grade	2pts
- Correctly identifies the second lowest grade	2pts
- Replaces lowest grade with second lowest grade successfully	4pts
- Returns correct average	2pts

cellphoneBill(10pts)

- Loops through input list	2pts
- Identifies which elements fall within which ranges	1pt
- Calculates correct charge for each element in the input list	2pts
- Correctly sums individual charges	1pt
- Applies discount correctly	2pts
- Returns correct total amount	2pts

strElongation (10pts)

- Loops through input string	2pts
- Identifies the amount each letter needs to be multiplied by	2pts
- Correctly adds multiple of each letter to output string	4pts
- Output string is correct	2pts

numHourglass (15pts)

- Correct number of rows	5pts
- Each row is the correct number	5pts
- Not hardcoded	5pts

gpaCalculator(20pts)

- Takes in two correctly formatted parameters	2pts
- Loops through both lists	2pts
- References correct elements in both lists when calculating qty. pts	4pts
- Correctly calculates quality points for each grade-hours pair	6pts
- Correctly calculates GPA	4pts
- Returns GPA in the correct format	2pts

nextRow(10pts)

- Returns correct nextRow for all possible input rows	10pts

turtleBattery(15pts)

- Takes in 2 parameters	1pt
- Long hand stays at 12	3pts
- Short hand is drawn at correct hour position	7pts
- Short hand is (recognizably) shorter than long hand	2pts
- Turtle is stamped at each hour position	2pts
- Turtles are stamped in the correct direction	3pts
- The clock size changes as radius changes	