CS 1301 Individual Homework 3
Conditionals & Loops

Due: **Friday, January 30th, 2015 before 11:55 pm**

Out of 100 points

File to submit: HW3.py

Students may only collaborate with fellow students currently taking CS 1301, 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.**

Functions

You will write a few python functions for practice with the language. In your HW3.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 input will be valid.

Function Name: **tallEnough (5 pts)**

Parameters:
inches - an integer representing the user's height in inches

Return Value:
Either “Have Fun!” or “Sorry, you can't play”
Example Cases:

```
python>>>tallEnough(48)
'Sorry, you can't play'
python>>> tallEnough(44)
'Have fun!'
python>>> tallEnough(46)
'Sorry, you can't play'
```

Description:

Write a function to determine if children are allowed on a playground. A child must be above 50 cm and under 1 meter 12 cm to play. If the user's height, which is provided by the parameter inches, meets these requirements, return the String “Have fun!otherwise return “Sorry, you can't play”
Use the conversion 1 cm = 1 inch / .39370

Function Name: **battleship (10 pts)**

Parameters:
Int1- the x coordinate of the ship's position
Int2- the y coordinate of the ship's position
*You may assume these positions are non-negative
Return Value:
None
Test Cases:

```
battleship(1,1)
input box: Guess the x-coordinate: 3
input box: Guess the y-coordinate: 1
You missed! Try again.
input box: Guess the x-coordinate: 1
input box: Guess the y-coordinate: 8
You missed! Try again.
input box: Guess the x-coordinate: 1
input box: Guess the y-coordinate: 1
You Hit!
```

Description:
Write a function to do a fun game of battleship. The function should take in two parameters: the x-coordinate and the y-coordinate of the ship's position. Then, the function asks for the user to input guesses separately for x and y coordinates. If the combination of these two is correct, then it should print a string that lets the user know they hit their mark. If it is wrong, the function should let the user know and ask again. The function should keep asking for coordinates until the user enters the correct ones! [Note that battleships are size 1x1.]

Function Name: **allNumbers (10 pts)**

Parameters:
inputString - A String.
Return:
A String.
Test Cases:

```
python>>> allNumbers("L33t H4X0R")
'3340'
python>>> allNumbers("Daft Punk is awesome.")
''
python>>> allNumbers("Acht8 6Seiben7*&^53*09")
'8675309'
```

Description:
Write a function that uses a for loop to create and return a new string that contains only the numbers in the original input. If the input string has no numbers, you must return an empty string.
You <u>MUST use a for loop</u> for this problem! *Hint: Remember the line "import string" and use the "in" check along with the "string.digits" constant to determine if each character is a number or not.*

Function Name: **incrementNumbers (15 pts)**
Parameter:

inputString- string that the user enters

Return Value:
 None
Test Cases:
python>>> incrementNumbers("I ate 3 pies yesterday.")
Input Box > 3
I ate 4 pies yesterday.
python>>> incrementNumbers("There are 366 days in a leap year.")
Input Box > 6
There are 377 days in a leap year.

Description:
 Write a function that takes one String as a parameter. The function should then ask the user to input a single positive integer (using the input box). If the number is in the string, the function then goes ahead and replaces it with that number incremented by one. The function should then print the modified string. If the number given by the input box is not in the String, the function should print the original String.

You only need to account for the user entering a single (positive) digit. You will have to play around with your data types. The str() and int() functions are very useful here.

Function Name: **countDown (15 pts)**
Parameters:

start- an integer marking the starting number
end- an integer marking the ending number
decrement- an integer marking the size of decrement
Return Value:
None
Test Cases:

```
python>>> countDown(9,0,3)
9
6
3
0
Done!

python>>> countDown(8,6,2)
8
6
Done!
```

Description:
Write a function that takes in three parameters, a starting number, an ending number and the decrement. The function <u>MUST use a while loop</u> to print starting from the starting number all the way to the ending number all in a new line using the decrement. After the ending number is reached the function should print "Done!" in a new line. The function should return None. You can assume that the user will always put a starting number larger than the ending number, and the ending number can be reached by decrementing the starting number.

Function Name:**numMountainRange (20pts)**

Parameter:
X (Integer): An integer that specifies the number of rows of the

mountain. You may assume the number is an integer between 2-9.
Return Values:
None
Description:
Write a function that takes in the number of rows of the mountain as a parameter. The function will then draw a number mountain on screen using the print function. See below in the test cases for clarification. DO NOT HARD CODE THE PRINTOUTS.

Test Cases:
You have X number of rows, but note that there are two 1s, four 2s, six 3s, eight 4s, etc.

```
python>>> numMountainRange(6)
     11
    2222
   333333
  44444444
 5555555555
666666666666
Ok
python>>> numMountainRange(3)
   11
  2222
333333
Ok
```

Function Name: **printFibonacci (25 pts)**
Parameters:
first an integer
second an integer

Return Value:
none

Your job is to write a function that takes in two parameters and prints out a Fibonacci sequence (<u>In the same line, separated by commas!</u>) using those parameters. A Fibonacci sequence is produced by adding the two preceding numbers together to produce the next integer. The two parameters the user inputs will be the first two numbers you add to start your sequence. The function should stop when the last number printed is more than 300.

Remember, the numbers must be printed in the same line with commas separating them. It's okay if the output wraps around to a new line. The key is that you print a single String.

You don't need to print the parameters as a part of your output. You can also assume that the user will always input at least 1 non-zero parameter.

Examples:

```
python>>> printFibonacci(1,9)
10,19,29,48,77,125,202,327
```

```
python>>> printFibonacci(2,3)
5,8,13,21,34,55,89,144,233,377
```

```
python>>> printFibonacci(1,1)
2,3,5,8,13,21,34,55,89,144,233,377
```

Grading Rubric

tallEnough 5 pts

- function takes in a height in inches	1
- function correctly converts to centimeter	2
- function returns correct Strings	2

battleship 10 pts

- function asks the user to input x and y coordinates	2
- function should take two parameters	2
- function continues until correct numbers inputted	4
- function returns the correct value (None)	2

allNumbers 10 pts

- uses a for loop	4
- returns correct output for any valid input	6

incrementNumbers 15 pts

- Correct header / one parameter	3
- function asks the user to input a number	2
- the function prints out the new phrase	2
- the function correctly increments numbers	8

countDown 15 pts

- function accepts three parameters as integers	3
- function uses a while loop	5
- function prints out the right results	5
- function prints out 泥 one!at the end	2

numMountainRange 20pts

- Correct number of rows and correct number in rows	10
- Correct shape (-5 if hard coded)	10

printFibonacci 25pts	
- function takes two parameters	5
- function prints a fibonacci sequence	5
- function stops after going past 300	5
- function prints with commas separating numbers	5
- function prints on one line	5

Elements of this homework created by Micah Terrell