More Loop Techniques and Concepts

Important Concepts
When using loops, the following are important concepts that can be useful.

1) Counter

A counter is an integer variable that we use to keep track of the number of times a particular event has happened. The most common thing to count is the number of times we went around a loop. This is what we are counting in the above example, because with each time around the loop we enter one mark, so in effect we are counting the number of marks. To use a counter may be different if you are using a for loop, or a while.

In a for loop, the counter is already involved:

Example #1

```
# loop1
# Displaying all the numbers between 1 and 10 with a for-loop

for i in range(1, 11):
       print(i)
```

In the above example, i is the counter. It counts (or controls) the number of times we go through the loop.

For a while loop, the setup of a counter is extremely important not to forget.

1. Give the counter an initial value of 0. (before the loop)
2. Each time you encounter what you want to count increase the counter by one (x = x + 1) or simply (x += 1)
3. Typically you display the final value outside the loop

Example #2

```
# loop2
# Finding how many marks earn an A
numA = 0
while True:
      mark = int(input("Enter a mark. <-1 to exit> "))
      if mark == -1: #exit the loop
                  break
      elif mark >= 80:  #must be an A
                  numA += 1
print("There were %i A's." %numA)
```

2) Accumulator

An accumulator (or sum) is similar to a counter. Instead of increasing the variable by one we can increase it by any number. An accumulator is typically used to add up a bunch of numbers, such as the sum of all the numbers from 1 to 100, or the total number of ages. (`total = total + num`) or (`total += num`)

Example #3

```
# loop4
# Find the sum of all squares from 1 to 100 using a for loop

total = 0

for count in range(1,11):
    total += count ** 2

print ("The sum is", total)
```

Example #4

```
# loop5
# exponential growth and decay
# If you invest $5000 at 6% interest for 5 years (compounded yearly)
# how much would you have at the end?

money = 5000

# if only 1 value is given within range
# then it is the exit, and the initial value of year is 0
for year in range(5):
      money *= 1.06 #or money += (money*.06)

print ("The sum is", round(money,2))
```

3 Working with Strings

Example #5

What about for strings? Pay careful attention to this one.

```
# loop6
# Using Loops with Strings
# This is a simple program that adds dots after each letter

name = input("Enter your name:")
newName = ""

for let in name:
    newName = newName + let + "."

print (newName)
```

This example points out two important things. First strings accumulate differently. Rather than starting at zero, they start as an empty string "". When we add on letters they simply make the string longer. Second, the for loop allows us to have the "loop control variable" BECOME each letter of a string.

4 **Working with Graphics**

Example #6

Create a gradient effect across the screen, starting at red and slowly changing to green.

```
# loop6
# Using Loops with Graphics
# Creates a gradient from red to green.

import pygame
pygame.init()
RED = (255, 0, 0)
BLACK = (0,0,0)
SIZE = (800, 600)
screen = pygame.display.set_mode(SIZE)
red = 255
for col in range(0, 700, 3):
   pygame.draw.rect(screen, (red, 255-red, 0), (0, col, 800, 3))
   red -= 1
pygame.display.flip()
pygame.time.wait(3000)
pygame.quit()
```

Notice there are two things happening in the loop, the rectangle of width 3 is moving down the screen with the use of the variable *col*. The variable *red* is counting down from 255 (remember colour is from 255 to 0) and will become one less with each new rectangle.

More Programming With Loops

1) Write a program that finds the sum of all the even numbers from 0 to 100.

2) Write a program that asks the user to input two numbers. Your program should find the sum of all the numbers in between, inclusive (what does this mean?).

3) Ask the user to input a number between 1 and 15. Output the product of all the numbers from 1 to that number.

4) Write a program that asks the user to input a number. Your program outputs the sum of the digits.

5) Write a program that asks the user to input a number. Your program outputs the product of the odd digits. eg: 1321 = 1 X 3 X 1 = 3

6) Write a program that asks for a string from the user. Output the first and last letters on one line, the second and second last letters on the second line, etc…

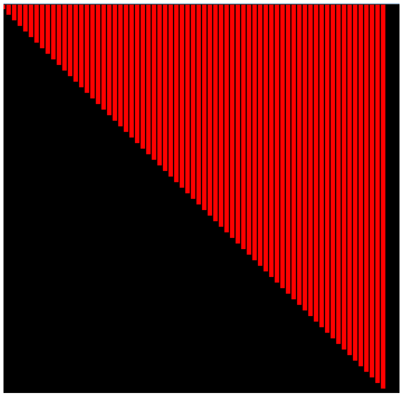
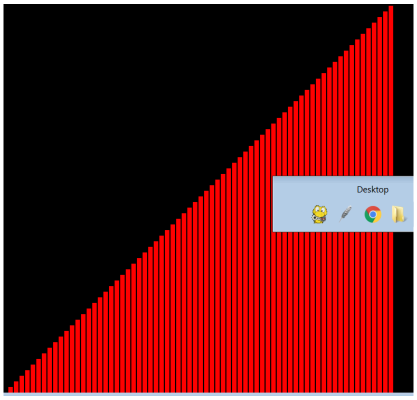
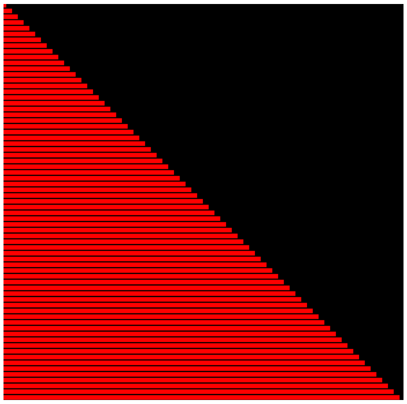
```
Example:    Please enter a string:   blah
            Output:
                  b     h
                  l     a
                  a     l
                  h     b
```

7) Write a program that asks the user to input a word. The program then rotates the characters of the word to create a square pattern of letters.

```
For example:  help
              elph
              lphe
              phel
```

8) Write a program that asks the user to input a sentence. Your program should then display the sentence with one word per line. You must use loops and not the replace command like we had earlier.

9) Each of the following columns are 10 pixels wide and multiples of 10 long on a 600 by 600 size screen. You can try to use a gradient as well. Random colours are fun too.

a)		**b)**	
c)		**d)**