New York University
School of Continuing and Professional Studies
Division of Programs in Information Technology

Introduction to Python
Homework, Session 2

2.1 Count from 0 up to 100 by 2's (or 3's, or 4's). Use a while True: loop to repetitively take one input from the user: an integer "step" value. Test to ensure that the value is an integer value, and allow the while True to repeat taking input until breaking out when correct. Then print out all the values from 0 up to 100 (or less than 100 if the step value doesn't divide evenly into 100).

Sample program run(s):

```
$ python myprog.py

please enter an integer:  x
sorry, 'x' is not a valid integer

please enter an integer:  3
counting from 0 to 100 by 3's


3
6
9
12

15  ... (etc., up to 99)

$ python myprog.py

please enter an integer: 25
counting from 0 to 100 by 25's

0
25
50
75
100
```

2.2 Sum a sequence of integers. As in previous, use a while True: loop to take a positive integer value (i.e., reject non-integers as well as values < 1. Now sum up all integers between 0 and the submitted value.

Sample program run(s):

```
$ python myprog.py
please enter a positive integer:  hey
sorry, 'hey' is not a valid positive integer
please enter a positive integer:  0
sorry, '0' is not a valid positive integer
please enter a positive integer:  1
sum from 1 to 1 is 1

$ python myprog.py
please enter a positive integer:  2
sum from 1 to 2 is 3

$ python myprog.py
please enter a positive integer:  3
sum from 1 to 3 is 6

$ python myprog.py
please enter a positive integer:  4
sum from 1 to 4 is 10

$ python myprog.py
please enter a positive integer:  5
sum from 1 to 5 is 15
```

2.3 Text replacement utility. Complete this program so that it can replace any word or phrase in a text file, and then output the file text after the replacements have been made. It can also report the number of replacements that were made.

Take two strings with two calls to input(): one for target text and the other for replace text. The supplied starter code will open and read the file out to a string; complete the program by replacing the target text with the replace text, and then report on the number of replacements made.

If you would like to apply this to a file, comment out the last line in this sample and uncomment the line before it. The input() statement prompts for the filename.

Sample program run(s):

```
python myprog.py

please enter search text:  yourself
please enter replace text:  your Corgi

And since you know you cannot see your Corgi,
so well as by reflection, I, your glass,
will modestly discover to your Corgi,
that of your Corgi which you yet know not of.

3 replacements made.
```

Starter Code:

```
sample_text = """And since you know you cannot see yourself,
so well as by reflection, I, your glass,
will modestly discover to yourself,
that of yourself which you yet know not of."""

def read_file(fname):
    try:
        text = open(fname).read()
    except IOError:
        print(('file {} not found or could not be read.  '.format(fname)))
        print('Using sample text instead.')
        text = sample_text
    return text

# to use a file instead of sample text, remove the comment mark (#)
# from the next line, and add a comment mark to the line after

# text = read_file(input('please enter a filename:  '))
text = sample_text

# [YOUR CODE GOES HERE]
```

EXTRA CREDIT HOMEWORK

2.4 Reverse Number Guessor. Create a program that selects a random number between 1 and 100 (see starter code below), but keeps it secret. It then tells the user whether his or her guess is correct, is greater than or is less than the user's guess.

Starter Code:

```
import random

number_to_guess = random.randint(1,100)
```

Sample program run(s):

```
I am thinking of a number from 1 to 100.   Try to guess it and I'll give you hints about it.

Your guess?  (q to quit):  50
your guess is LOWER than the number I'm thinking of.

Your guess?  (q to quit):  75
your guess is LOWER than the number I'm thinking of.

Your guess?  (q to quit):  87
your guess is HIGHER than the number I'm thinking of.

Your guess?  (q to quit):  81
you got it!  You guessed it in 4 tries.
```

2.5 Number Guessor. Write a program that attempts to guess a number from 1 to 100 that the user has chosen and is keeping secret. To guess the number, the program asks the user a series of guesses and questions, narrowing down the possibilities by half each time. The program will:

a. Ask the user to think of a number from 1 to 100
b. Ask the user if the number is 50 (i.e., halfway between 1 and 100
c. If the user answers "no", ask the user if the number is higher or lower than 50
d. If lower, ask the user if the number is 25 (i.e., halfway between 1 and 50)
e. If higher, ask the user if the number is 75 (i.e., halfway between 50 and 100)
f. Return to question c -- if the guess is incorrect, ask the user if the number is higher or lower than 25 (or 75)
g. Continue halving the possibiltiies and guessing the number until the user reports that the guess is correct.

Sample program run(s):

```
Think of a number between 100 and 0, and I will try to guess it.  Hit [Enter] to start.
is it 50 (yes/no/quit)?  no
is it higher or lower than 50?  higher
is it 76 (yes/no/quit)?  no
is it higher or lower than 76?  lower
is it 63 (yes/no/quit)?  no
is it higher or lower than 63?  lower
is it 57 (yes/no/quit)?  no
is it higher or lower than 57?  lower
is it 54 (yes/no/quit)?  no
is it higher or lower than 54?  lower
is it 52 (yes/no/quit)?  no
is it higher or lower than 52?  lower
is it 51 (yes/no/quit)?  yes
I knew I could do it!
```

2.6 Find prime numbers in a range of numbers. Ask the user for an integer, and display all prime numbers between 2 and the user's number.

Sample program run(s):

```
$ python primes.py

* Prime Numbers *
Please enter max integer:  5

2
3
5


$ python primes.py

* Prime Numbers *
Please enter max integer:  20

2
3
5
7
11
13
19
```

2.7 Chicken, Fox and Grain game. This is rather lengthy assignment using while, break and continue, as well as numerous if tests comparing strings and numbers. (In fact, it requires a great deal more if/then/else than might be needed if we were to use some upcoming Python features. But, it provides an interesting challenge using this week's features.)

The program is an implementation of a well-known logic puzzle:

A man has to transport a fox, a chicken and a sack of corn across a river using a rowboat, but the rowboat has room for only one of the items besides himself. Therefore when crossing the river he usually has to leave two of the items alone on one of the shores. The problem is that if he leaves the fox alone with the chicken, the fox will eat the chicken, and if he leaves the chicken alone with the grain, the chicken will eat the grain. How can he move all three items across the river without losing any of them?

Sample program run(s):

```
$ python myprog.py

**THE FOX, THE CHICKEN AND THE GRAIN**

You are on the east bank of a river.
You must transport a fox, a chicken
and a bag of grain to the west bank.
However, you can only carry one item
at a time.  Good luck!

you now are on the east bank.
the chicken is on the east bank.
the fox is on the east bank.
the grain is on the east bank.

What would you carry in the seat of the boat ([Enter] for nothing, q to quit)?  xx

sorry, I don't recognize 'xx'.  Try again.

you now are on the east bank.
the chicken is on the east bank.
the fox is on the east bank.
the grain is on the east bank.

What would you carry in the seat of the boat ([Enter] for nothing, q to quit)?

What would you carry in the seat of the boat ([Enter] for nothing, q to quit)?  grain

oops, the fox got the chicken!


$ python myprog.py

**THE FOX, THE CHICKEN AND THE GRAIN**

You are on the east bank of a river.
You must transport a fox, a chicken
and a bag of grain to the west bank.
However, you can only carry one item
at a time.  Good luck!

you now are on the east bank.
the chicken is on the east bank.
the fox is on the east bank.
the grain is on the east bank.

What would you carry in the seat of the boat ([Enter] for nothing, q to quit)?  fox

oops, the chicken got the grain!


$ python myprog.py


**THE FOX, THE CHICKEN AND THE GRAIN**

You are on the east bank of a river.
You must transport a fox, a chicken
and a bag of grain to the west bank.
However, you can only carry one item
at a time.  Good luck!

you now are on the east bank.
```

```
the chicken is on the east bank.
the fox is on the east bank.
the grain is on the east bank.

What would you carry in the seat of the boat ([Enter] for nothing, q to quit)?  chicken

...carrying the chicken to the west bank...

you now are on the west bank.
the chicken is on the west bank.
the fox is on the east bank.
the grain is on the east bank.

What would you carry in the seat of the boat ([Enter] for nothing, q to quit)?
[pressed Enter, meaning take back nothing]

...traveling to the east bank...

you now are on the east bank.
the chicken is on the west bank.
the fox is on the east bank.
the grain is on the east bank.

What would you carry in the seat of the boat ([Enter] for nothing, q to quit)?  fox

...carrying the fox to the west bank...

you now are on the west bank.
the chicken is on the west bank.
the fox is the west bank.
the grain is on the east bank.

What would you carry in the seat of the boat ([Enter] for nothing, q to quit)?
[pressed Enter, meaning take back nothing]

oops, the fox got the chicken!
```