CMPS 5P
Introduction to Programming in Python
Programming Assignment 5

In this assignment you will write a Python program that plays the other side of the guessing game you implemented in pa3. Your program will ask the user for an initial search space by prompting for two integers giving the lower and upper limits respectively. If these two integers are out of order (i.e. low > high) it will continue to prompt for appropriate limits. (It is not necessary to recover from non-integer input at this stage.) Your program will then ask the user to think of a number in the given range. It will make a sequence of guesses that emulate the Binary Search algorithm discussed in class. (See the example BinarySearch.py posted on the class webpage.) It will ask the user if the mystery number is Less than, Greater than or Equal to each guess. Subsequent guesses will depend on how the user answers these queries. If at some point the user acknowledges that the guess is equal to the mystery number, your program will state the number of guesses used. If at some point the current search space consists of just one number your program will stop guessing, announce the number and state the number of guesses used. If at some point the search space becomes empty, your program will state that the answers given by the user were inconsistent.

Your program will be called Question.py. A typical session with the user is reproduced below.

```
$ python Question.py


Enter two numbers, low then high.
low = 10
high = 1

Please enter the smaller followed by the larger number.
low = 5
high = 3

Please enter the smaller followed by the larger number.
low = 1
high = 10

Think of a number in the range 1 to 10.

Is your number Less than, Greater than, or Equal to 5?
Type ’L’, ’G’ or ’E’: x

Please type ’L’, ’G’ or ’E’: y

Please type ’L’, ’G’ or ’E’: e

I found your number in 1 guess.


$
```

Observe how the user is prompted repeatedly if the limits are out of order. Notice also the blank lines that appear before and after user input. Your program will accept only the letters L, G, or E (upper or lower case) as valid input, and continues to prompt if any other input is supplied. Two more sessions follow.

```
$ python Question.py


Enter two numbers, low then high.
low = 2
high = 18

Think of a number in the range 2 to 18.

Is your number Less than, Greater than, or Equal to 10?
Type ’L’, ’G’ or ’E’: g

Is your number Less than, Greater than, or Equal to 14?
Type ’L’, ’G’ or ’E’: L

Is your number Less than, Greater than, or Equal to 12?
Type ’L’, ’G’ or ’E’: L

Your number is 11. I found it in 3 guesses.


$ python Question.py


Enter two numbers, low then high.
low = 1
high = 10

Think of a number in the range 1 to 10.

Is your number Less than, Greater than, or Equal to 5?
Type ’L’, ’G’ or ’E’: L

Is your number Less than, Greater than, or Equal to 2?
Type ’L’, ’G’ or ’E’: g

Is your number Less than, Greater than, or Equal to 3?
Type ’L’, ’G’ or ’E’: l

Your answers have not been consistent.


$
```

Together these examples illustrate all the ways the game can end. Either the user acknowledges that the number was guessed by entering 'E', or the search space narrows to one number and the program announces this fact, or the program notices that the users responses are not logically consistent.

To get full credit your output should be formatted exactly as above, down to details like punctuation and pluralization of the word "guess" (i.e. "1 guess" vs. "2 guesses".) The final session below shows that it is possible for the program to end with no guesses.

```
$ python Question.py


Enter two numbers, low then high.
low = 7
high = 7

Think of a number in the range 7 to 7.

Your number is 7. I found it in 0 guesses.


$
```

What to turn in

Submit the file Question.py to the assignment name pa5 in the usual way. As always, start early and ask questions if anything is not clear.