22C:16 Practice Problem Set 4

Morning Section: Complete before Tuesday, 2-19-2013

Evening Section: Complete before Monday, 2-18-2013

These practice problems correspond roughly to the material covered in the fourth week of classes (2-11 to 2-15).

1. Consider the following expressions. For each expression, first determine whether the expression is *well-formed* or not. For each well-formed expression, determine if the expression will be successfully evaluated by Python or not. For each expression that is successfully evaluated, determine its value and the type of its value.

   You should ideally work through each expression away from a computer and then compare your answers with what you get by using the Python shell. Make sure that understand the reason behind each answer.

   (a) `3 < 10 or 5`

   (b) `3 < (10 or 2)`

   (c) `3 < (2 or 10)`

   (d) `(10 < 100) or (50 < 100/0)`

   (e) `(50 < 100/0) or (10 < 100)`

   (f) `len("hello") == 25/5 or 20/10`

   (g) `-4*-2**3+-len(str(6))`

   (h) `2*(2*(2*len("01"))`

   (i) `(5 < 10) and (10 < 5) or (3 < 18) and not 8 < 18`

   (j) `5 < -20`

   (k) `5 < *20`

   (l) `"expression"/10 + 15`

   (m) `(len("expression")/10 + 15)/len(str(10+15))`

   (n) `round(10.78)/3 < len("10.78)`

   (o) `abs(5 - 25) < 15`

   (p) `len(bin(25))`

   (q) `bool(math.sqrt(-1))`

   (r) `bool(math.sqrt((-1)**10))`

   (s) `math.ceil(1.5)/math.floor(1.5)+200L`

   (t) `(sys.maxint + 10)*0`

   (u) `2/math.trunc(1.5)+200L`

   (v) `sys.max+10`

   (w) `10**350`

   (x) `10.0**350`

   (y) `random.randint(0, 20) <= 100`

2. Write down the value and type of each of these expressions. Assume that the `math` and the `sys` modules have been imported prior to the execution of these expressions. Also, suppose that the value of `sys.maxint` is `9223372036854775807`.

   (a) `100L + 200`

   (b) `math.ceil(10.97) - math.floor(11.17)`

   (c) `len(str(10.97))`

   (d) `bin(5) + bin(3)`

   (e) `sys.maxint + 2`

3. For each of the following expressions, first determine if evaluating it would cause Python to report an error; if so, explain in a sentence what it was about the expression that caused the error. Otherwise, if you think the expression will evaluate in Python in an error-free manner, write down the *value* that the expression evaluates to and its *type*. Assume that the modules `math`, `sys`, and `random` have already been imported prior to the evaluation of these expressions.

   (a) `bin(len(str(10<20)))`

   (b) `4L*len("Problem1")/(len("Exam1")%3)+2.0`

   (c) `len('twelve'+"four")/(len('2L')+2*(-1)**3)`

   (d) `round(math.pi, 2) < math.trunc(math.pi)`

   (e) `20 < 30 and 50 < 25 and 4 < 100/0`

   (f) `str(12 + 4.0/2**2)`

   (g) `3.0 * input("Enter an integer:  ")`
   (Assume that user inputs the expression `10/2 + 1`, when prompted)

   (h) `not bool(0.1) or not (100 and not False)`

   (i) `2*float("sys.maxint")`

   (j) `len(str(random.randint(1, 4)*22))`

4. For each of the programs below, write down the output produced by the program.

(a) What output does this program produce?

```
x = 10
while x < 120:
    y = x + 40
    while (x < y) :
        if (y % 10) == 5 :
            y = y + 15
        else :
            print x, y
            y = y - 35
    x = x + 30
```

(b) What output does this program produce?

```
x = 0
while x < 10:
    if x % 2 == 0:
        y = x + 1
        while (y < 11) :
            print "Line 1:", x, y
            y = y + 4
    else :
        y = 11-x
        while (y > 1) :
            print "Line 2:", x, y
            y = y - 3
    x = x + 3
```

5. Here is a partially completed program that aims to solve the following problem. The user types in a sequence of positive integers, one per line, ending with the number `0`. The program reads in this sequence and counts and outputs the number of *pairs of consecutive numbers that are in increasing order*. An example interaction of this program with the user is given below.

```
Type a positive int (zero if done).20
Type a positive int (zero if done).23
Type a positive int (zero if done).25
Type a positive int (zero if done).20
Type a positive int (zero if done).19
Type a positive int (zero if done).9
Type a positive int (zero if done).10
Type a positive int (zero if done).0
3
```

The program outputs 3 because it detects three pairs of consecutive numbers in increasing order: (i) $20, 23$, (ii) $23, 25$, and (iii) $9, 10$. The program below has two blanks to fill. Your task is to fill in these blanks.

```
import sys

# Variable used to read input numbers
current = int(raw_input("Type a positive int (zero if done)."))

# tracks the number just prior to the most recently read number
previous = sys.maxint

# counter to track the number of consecutive, increasing pairs
numIncreasingPairs = 0

while current:
    if current > previous:
        numIncreasingPairs = numIncreasingPairs + 1

    # Update previous (Blank 1)

    ------------------------------------------

    # Update current (Blank 2)

    ------------------------------------------------------

print numIncreasingPairs
```

6. I have defined a function called `startsWith` whose header is given below:

```
def startsWith(word, letter):
```

Both parameters `word` and `letter` are expected to be of string type and furthermore the second parameter, i.e., `letter` is expected to be of length 1. The function returns `True` if the first character in `word` happens to be equal to `letter`; otherwise the function returns `False`. For example, the function call `startsWith("hello", "x")` returns `False` and the function call `startsWith("fellow", "f")` returns `True`.

(a) Define a function `startsWithDigit` that takes a single parameter `word` and returns `True` if the given `word` starts with "0", "1", "2", ..., "9". The function `startsWithDigit` should repeatedly and appropriately call the given function `startsWith` to complete its task.

(b) Write a program that prompts a user for their "password" and then determines if the password is *valid* or not. A password is said to be *valid* if it starts with a digit **and** it has length 6 or more. If your program determines that the user-entered password is not valid, it should print a message saying so. Otherwise, it should print a message saying that it has accepted the user-entered password. Your program should make use of the function `startsWithDigit` defined above.
