Validating / debugging code

In these problems, the code is given and the aim is to test code comprehension at Schulte's *functional* level, that is, working out how the code given relates to a specified task. The formats here could be:

The following code is an attempt to solve <this problem>. Determine whether it works correctly. If it does not, suggest a correction.

or, as we discussed today, the (probably) slightly easier variant:

The following code is a solution to <this problem>. It contains errors. Find the errors and suggest appropriate corrections.

For both of these examples, some sample input can optionally be provided.

This format allows the students to see and work with code that solves, or nearly solves, specified problems - hence increasing their exposure to possible code patterns.

---

1. Tricky Twelve Times Tables!
The following code attempts to print out the 12 times table, from 1x12=12 to 12x12=144 inclusive - however there are three errors. Can you find them and fix them?

```
for i in range (1,12):
      print("ix12=", (12i))
```

2. Logical Lights
The security light should be turned on if it is dark and there is motion detected. These are three boolean variables `lightOn`, `isDark` and `motionDetected`.

What's wrong with this assignment statement?

```
lightOn = isDark or motionDetected
```

More Logical Lights
The light sensor is re-programmed so the light is switched on if it is dark and before midnight, or it is dark and motion is detected. Given the boolean variables `lightOn`, `isDark`, `motionDetected` and `beforeMidnight` - what's wrong with this assignment statement?

```
lightOn = (isDark and motionDetected and beforeMidnight)
```

3. Loopy Navigation
The following code asks "are we there yet?" and waits for input from the user . The code should ask the question again unless the answer is "yes". Can you find two problems in the program?

```
answer = "no"
while answer == "yes":
     input("are we there yet? ")
print("we have arrived at last!")
```

4. Spot that Leap Year
A year is a leap year if it is a multiple of 4, so long as it is not a multiple of 400. So 1996 was a leap year, but 2000 wasn't. Check this leap year logic below, and spot the three mistakes

```
def leapYear(year):
    if year % 4 == 0 and year % 400 == 0:
    return True
```

5. The Longest List
If `shopping` is a list of lists, like this:

```
shopping = [ ['eggs', 'ham'],
             ['bread', 'cheese', 'butter', 'branstonpickle'],
             ['toothpaste', 'toothbrush'],
             ['fish', 'chips', 'mushy peas'] ]
```

The following code is meant to find the longest list inside `shopping`, and print out all the items in this list. Find and correct the mistake(s).

```
max = 0
for l in shopping:
     if len(l) > max:
     max = len(l)
print('the longest list is ')
for item in shopping[max]:
     print(item)
```

6. If I were you…
What’s wrong with this conditional code, which checks the user’s name?

```
name = input(‘what’s your name?’ )
if name = ‘jeremy’:
print(‘what a great name!)
else
print(‘if only your name was jeremy!’)
```

7. Dodgy Dictionaries
Suppose you have a dictionary named roman that has keys which are roman numerals represented as strings, and values that are the equivalent integers. E.g:

```
roman[‘i’]  = 1
roman[‘ii’]  = 2
roman[‘v’]  = 5
```

Here is a program that reads in two roman numeral values, adds together the corresponding integer values and prints out the answer as an integer. Can you find the mistakes?

```
first = input(‘enter a roman numeral’)
second = input(‘and another roman numeral’)
answer = roman(first) + roman(second)
print(‘the answer is’ + answer)
```

8. Correct Change
This is some code from a Supermarket Automated Checkout. The following function `giveChange` takes two parameters - `amount` is the amount of change (in pence) to be given, and `coins` is the list of values of coins (also in pence, sorted from highest to lowest value) that are available. The function giveChange should return a list of coins that make up the change amount. For instance:

`giveChange(23, [50, 20, 10, 5, 2, 1])` should return `[20, 2, 1]`
`giveChange(99, [50,20,10,5,2,1])` should return `[50,20,20,5,2,2]`

There are two mistakes in this code - can you spot them?

```
def giveChange(amount, coins):
      change = []
      for c in coins:
      while amount > c:
            amount = amount-c
            change = change + [c]
      return change
```

9. Superhero Stars

Given a file `reviews.txt`, which has the following lines:

```
Spiderman ****
Batman *****
Wonderwoman **
Superman *
Hulk ***
```

Inspect the following code, which opens the file and finds out which superhero has the highest star rating (it should be Batman, with 5 stars). The code is meant to print out the name of the superhero with the highest rating, and how many stars s/he has. Can you find three problems with this code?

```
f = open('reviews.txt', 'r')
max = 0
n = 0
superhero = ''
for line in f:
     if '*' in line:
     hero=(line.split())[0]
     stars=(line.split())[1]
          for c in stars:
          if c=='*':
               n = n+1
               if n > max:
                    max = n
                    superhero = hero
print(superhero + ' has ' + int(max) + ' stars')
```