Lecture 6 - Exercises

6A - Conditional Statements

1. What does the following program print?

   ```
   wwu_founded = 1893
   if (wwu_founded // 1000) < 1:
       wwu_founded = "eighteen ninety three"
   if type(wwu_founded) == type("some text"):
       print("WWU was founded in", wwu_founded)
   else:
       print("Year founded:", wwu_founded)
   ```

2. Some of the following expressions could fill in the blank below to make the following program print `oolong`. Which ones are they?

   ```
   if (True and (_______ or not True)) and not False:
       print("green")
   else:
       print("oolong")
   ```

   1. `True`
   2. `False`
   3. `not True`
   4. `not False`
   5. `3 == 4`
   6. `17 % 2 == 1`

3. Is there anything that could fill in the blank in the program above that would cause the program to print both `green` and `oolong`?

6B - Nested and Chained Conditionals

4. Consider the following program.

   ```
   # assume x and y are numbers
   if x < y:
       print("x is less than y")
   else:
       if x > y:
           print("x is greater than y")
       else:
           print("x and y must be equal")
   ```

   1. How many comparison operators (`>`, `<`) are evaluated when the program is run with `x` set to `4` and `y` set to `5`?
   2. How many comparison operators (`>`, `<`) are evaluated when the program is run with `x` set to `8` and `y` set to `3`?
   3. Rewrite the program to use chained conditionals (i.e., `elif`) intstead of nested `if` statements.

5. Consider the following program:

   ```
   # Program 0:
   if (a < 0) == True:
       print(0)
   else:
       if a >= 0:
           print(a)
   ```

   1. Which of the following programs are equivalent? In other words, which of the following has exactly the same output regardless of the value of `a`?

```
# Program 1:
if a < 0:
    print(0)
else:
    print(a)

# Program 2:
if (a < 0) == True:
    print(0)
print(a)

# Program 3:
if (a > 0) == True:
    print(a)
else:
    print(0)

# Program 4:
if (a < 0) == True:
    print(0)
if a >= 0:
    print(a)
```

2. Of the programs above that are equivalent to 0 (including Program 0 itself), which is the simplest and thus most preferable way to write it?

6. Consider the following three programs. Give the **smallest positive integer** value for the variable `num_tacos` such that the three programs print **exactly** the same thing when they are executed.

Program 1:

```
if (num_tacos == 32):
    print("32 tacos")
elif (num_tacos < 32):
    print("Too few tacos")
elif (num_tacos == 32):
    print("32 tacos")
elif (num_tacos % 5 == 0):
    print("Oh yes, tacos!")
else:
    print("Too many tacos")
```

Program 2:

```
if (num_tacos == 32):
    print("32 tacos")
if (num_tacos < 32):
    print("Too few tacos")
if (num_tacos == 33):
    print("33 tacos")
if (num_tacos % 5 == 0):
    print("Oh yes, tacos!")
else:
    print ("Too many tacos")
```

Program 3:

```
if (num_tacos == 32):
    print("32 tacos")
else:
    if (num_tacos < 32):
        print("Too few tacos")
    else:
        if (num_tacos == 34):
          print("34 tacos")
        else:
            if (num_tacos % 5 == 0):
                print("Oh yes, tacos!")
            else:
                print ("Too many tacos")
```

Problems

1. We made it! We finally know how to give the user of our math quiz program the customized feedback they deserve. Modify (or rewrite) your program from E05 Problem 1 to behave as follows:

   ```
   What is 4 * 6? 24
   Your answer is correct!
   ```

   ```
   What is 4 * 6? 18
   Incorrect - better luck next time.
   ```

2. The time of day can be written using clock time with an “AM” or “PM”, or it can be written using 24-hour time (sometimes called “military time”). In 24-hour time, minutes are written the same, but the hour of the day ranges from 0 to 23, so you don’t have to specify AM or PM. Hours from 1 to 12 are written the same - 1:00 AM is 1:00, 10:00 AM is 10:00. Hours after noon continue counting from 12 instead of going back to 1, so 1:00 PM is written 13:00, and 8PM is written 20:00. Midnight is written 0:00, and 11:59PM is written 23:59.

   Write a program that prompts a user for an hour, minute, and AM or PM, then prints the given time in 24-hour format.

```
Enter the hour: 10
Enter the minute: 0
Enter AM or PM: AM
10:00
```

```
Enter the hour: 10
Enter the minute: 35
Enter AM or PM: PM
22:35
```

Test your program thoroughly - make sure it works for “edge cases”, like midnight and noon.

3. Write a program that promtps a user for the start and end time of an event and tells them whether the event conflicts with the CSCI 141 class time (just lecture, not lab). To make things simpler for ourselves, let’s consider hours only and assume the user enters the hour in 24-hour time. You can also assume that no event starts before midnight and ends after midnight.

   ```
   What day is the event? Monday
   What time does the event start? 8
   What time does the event end? 10
   That doesn't conflict with your 141 lecture. Have fun!
   ```

   Note: The following assumes you’re in the 11am lecture section; the result for the 1pm lecture section would be different!

   ```
   What day is the event? Wednesday
   What time does the event start? 10
   What time does the event end? 12
   That event conflicts with your 141 lecture. Too bad you can't go!
   ```

4. Recall the `xor` ("exclusive or") operator from last time: `a` XOR `b` is true if exactly one of `a` and `b` is true, and false otherwise. Without using the `and`, `or`, and `not` operators, write a program that prompts a user for boolean values `a` and `b` and prints the value of `a` XOR `b`. *Hint 1:* The `bool` type conversion function will convert an integer to a boolean; 0 is converted to False, and 1 is converted to True. *Hint 2:* You'll probably need nested conditionals!

```
Enter a (0 for False, 1 for True): 1
Enter b (0 for false, 1 for True): 0
a XOR b is True
```

```
Enter a (0 for False, 1 for True): 1
Enter b (0 for false, 1 for True): 1
a XOR b is False
```