

COSC 1336 Lab: 6
Relevant reading: Chapter 5
Due: Oct. 23, 2:30 pm
(Late date: Oct. 30, 2:30 pm)
50 Points

Input validation

Problem 1. [10 points] Recall the `if/elif` version of the color mixer program from lab 4 (problem 3.b). In a file called `color_mixer_validation.py`, write a new version of the program, either by starting from scratch or by modifying your previous program. The new program should do input validation on the user's input when the two inputs are read in, to ensure that each is a valid primary color. This should simplify the logic of the rest of the program a bit (and you should make the appropriate simplifications). After the input validation loop(s), you might consider adding the following reasoning to your code so that you can further simplify the logic in the rest of the program:

```
# if the colors are not alphabetical...
if color1 > color2:
    # swap them so that they are
    color1, color2 = color2, color1
```

If you were not able to produce a successful solution to problem 3.b on lab 4 and you are still unsure about how to do so, you should be sure to talk to me during lab or office hours.

Nested loops

Problem 2. [12 points] Download the `magic_dates_validation_sentinel.py` file from Blackboard. In it, you will find a version of the magic dates program from lab 3 (problem 3) that has input validation added to it, to be sure that the date entered by the user is a valid date. For this problem, modify the program so that it uses a sentinel loop to allow the user to enter multiple dates. The user should indicate that he or she is finished entering data by entering 0 for the month. Be sure to update prompts to the user as appropriate. The following transcript shows what the program should look like when it is run.

Problem 3. [14 points] Make a new file called `ascii_triangles.py`.

- a. **[10 points]** Write a function called `print_triangle` that uses nested loops to draw this pattern:

```
#
##
# #
#  #
#   #
#    #
#####
```

- b. [4 points] Write another function called `print_triangle_with_height` that prompts the user to enter the height of the triangle. The function should use input validation to ensure that the height is at least 3 and no more than 200. Then it should draw the same style of triangle as in part a, but with the specified height. Consider starting with a copy of `print_triangle` and modifying it.

Problem 4. [14 points] The following problem is problem #9 of the Project Euler puzzles¹. For this lab, use nested loops to solve the puzzle and display only the solution to the problem. After the problem statement, there are some hints that you might find useful, but you may want to give the problem a shot without looking at the hints first. Put your program in a file called `pythagorean_triplet.py`

A Pythagorean triplet is a set of three natural numbers, $a < b < c$, for which,

$$a^2 + b^2 = c^2$$

For example, $3^2 + 4^2 = 9 + 16 = 25 = 5^2$.

There exists exactly one Pythagorean triplet for which $a + b + c = 1000$.

Find the product abc .

Here are some hints that may help you.

- You will need to consider negative natural numbers in your search.
- If you have two integer variables `a` and `b` from the formula above, you can calculate `c` in this way:
`c = sqrt(a*a + b*b)`
- If you have a float variable `c` and you want to know whether it contains an integral value, you can compare `c` to `int(c)`. If they are the same, then `c` contains an integral value.
- Consider having the outer loop increment `a` each time through, while the inner loop increments `b`. Remember that $a < b$.

For this lab, you should submit the following files:

- `color_mixer_validation.py`
- `magic_dates_validation_sentinel.py`
- `ascii_triangles.py`
- `pythagorean_triplets.py`

¹See www.projecteuler.net for more similar puzzles if you find such things entertaining.